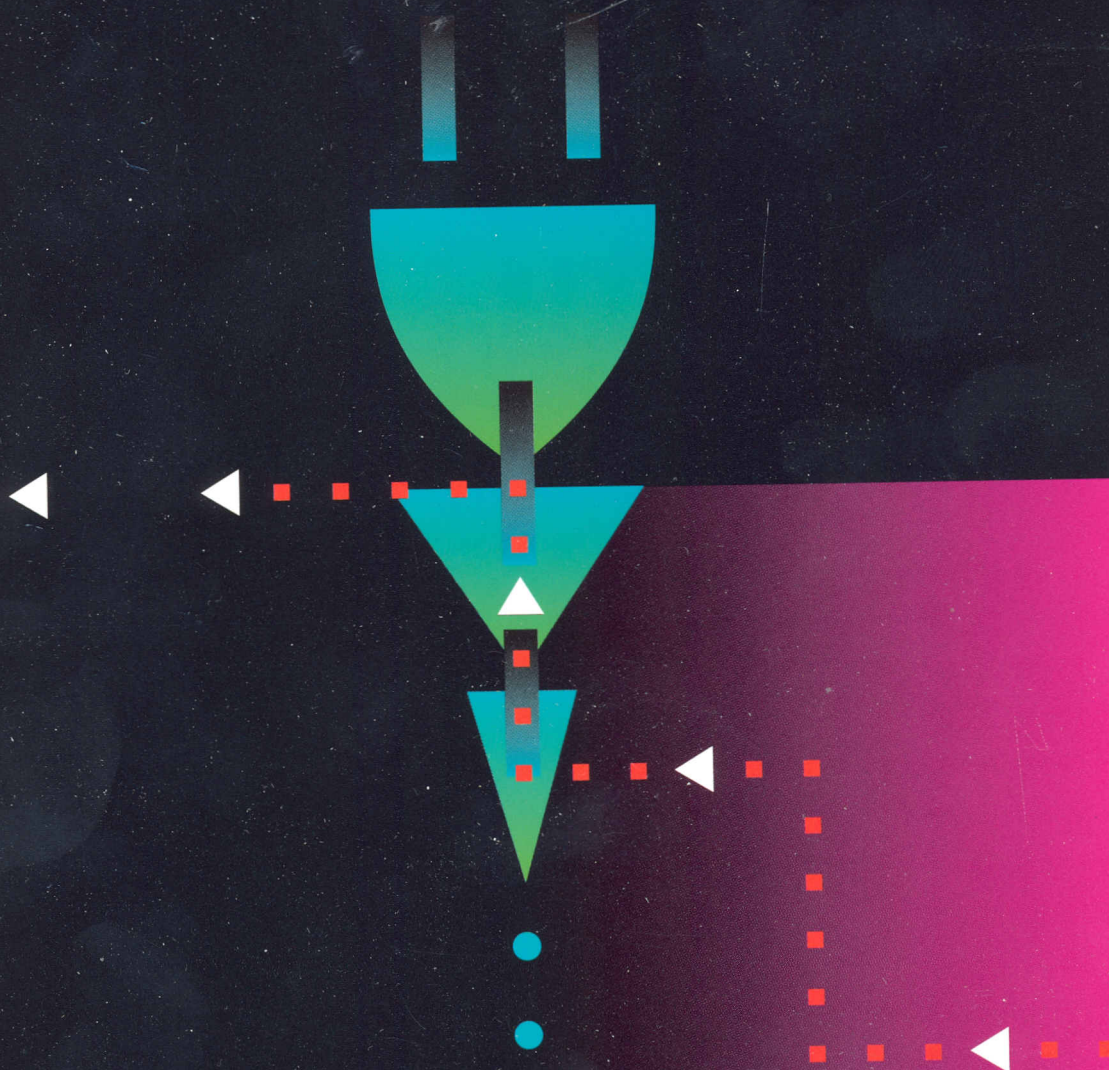


# Programmable Logic



intel®





## LITERATURE

To order Intel literature or obtain literature pricing information in the U.S. and Canada call or write Intel Literature Sales. In Europe and other international locations, please contact your **local** sales office or distributor.

**INTEL LITERATURE SALES**  
P.O. Box 7641  
Mt. Prospect, IL 60056-7641

**In the U.S. and Canada**  
call toll free  
(800) 548-4725

*This 800 number is for external customers only.*

### CURRENT HANDBOOKS

Product line handbooks contain data sheets, application notes, article reprints and other design information. All handbooks can be ordered individually, and most are available in a pre-packaged set in the U.S. and Canada.

Title	Intel Order Number	ISBN
<b>SET OF TWELVE HANDBOOKS</b> (Available in U.S. and Canada)	<b>231003</b>	<b>N/A</b>
<b>CONTENTS LISTED BELOW FOR INDIVIDUAL ORDERING:</b>		
<b>CONNECTIVITY</b>	231658	1-55512-174-8
<b>EMBEDDED APPLICATIONS (1993/94)</b>	270648	1-55512-179-9
<b>EMBEDDED MICROCONTROLLERS &amp; PROCESSORS</b> (2 volume set)	270645	1-55512-176-4
<b>MEMORY PRODUCTS</b>	210830	1-55512-172-1
<b>MICROCOMPUTER PRODUCTS</b>	280407	1-55512-173-X
<b>MICROPROCESSORS (2 volume set)</b>	230843	1-55512-169-1
<b>MOBILE COMPUTER PRODUCTS</b>	241420	1-55512-186-1
<b>i750®, i860™, i960® PROCESSORS AND RELATED PRODUCTS</b>	272084	1-55512-185-3
<b>PACKAGING</b>	240800	1-55512-182-9
<b>PERIPHERAL COMPONENTS</b>	296467	1-55512-181-0
<b>PRODUCT OVERVIEW</b> (A guide to Intel Architectures and Applications)	210846	N/A
<b>PROGRAMMABLE LOGIC</b>	296083	1-55512-180-2
<b>ADDITIONAL LITERATURE:</b> (Not included in handbook set)		
<b>AUTOMOTIVE</b>	231792	1-55512-125-X
<b>COMPONENTS QUALITY/RELIABILITY</b>	210997	1-55512-132-2
<b>CUSTOMER LITERATURE GUIDE</b>	210620	N/A
<b>INTERNATIONAL LITERATURE GUIDE</b> (Available in Europe only)	E00029	N/A
<b>MILITARY AND SPECIAL PRODUCTS (2 volume set)</b>	210461	1-55512-189-6
<b>SYSTEMS QUALITY/RELIABILITY</b>	231762	1-55512-046-6
<b>HANDBOOK DIRECTORY</b> (Index of all data sheets contained in the handbooks)	241197	N/A

LITINCOV/103092



## Programmable Logic

Programmable logic devices (PLDs) are used to develop smaller systems, reduce system costs and obtain higher performance while maintaining design security and increased flexibility.

Intel's programmable logic devices provide a complete range of design entry, simulation, compilation and programming tools. Intel PLDs – based on erasable programmable read-only memory for instant reconfiguration – offer designers the ability to customize their own system logic by tailoring the devices to meet performance requirements.

Intel offers PLDs with high speed, low power consumption and a range of integration capabilities. They are ideally suited for high-performance microcomputer system applications.

This handbook contains data sheets, application notes and technical briefs for Intel's PLD family, including information on PLDshell Plus, the design software program used to design, compile and program Intel's PLDs.



Order Number: 296083-008  
ISBN: 1-55512-180-2  
Printed in USA/193/30K/RRD DG  
Programmable Logic



Printed on  
Recycled Material







---

Founded in 1968 to pursue the integration of large numbers of transistors onto tiny silicon chips, Intel's history has been marked by a remarkable number of scientific breakthroughs and innovations. In 1971, Intel introduced the 4004, the first microprocessor. Containing 2300 transistors, this first commercially-available computer on a chip is considered primitive compared with today's million-plus transistor products.

Innovations such as the microprocessor, the erasable programmable read-only memory (EPROM) and the dynamic random access memory (DRAM) revolutionized electronics by making integrated circuits the mainstay of both consumer and business computing products.

Over the last two and a half decades, Intel's business has evolved and today the company's focus is on delivering an extensive line of component, module and system-level building block products to the computer industry. The company's product line covers a broad spectrum, and includes microprocessors, flash memory, microcontrollers, a broad line of PC enhancement and local area network products, multimedia technology products, and massively parallel supercomputers. Intel's 32-bit X86 architecture, represented by the Intel386™ and Intel486™ microprocessor families, are the de facto standard of modern business computing and installed in millions of PCs worldwide.

Intel has over 25,000 employees located in offices and manufacturing facilities around the world. Today, Intel is the largest semiconductor company in the United States and the second largest in the world.









## U.S. and CANADA LITERATURE ORDER FORM

NAME: \_\_\_\_\_

COMPANY: \_\_\_\_\_

ADDRESS: \_\_\_\_\_

CITY: \_\_\_\_\_ STATE: \_\_\_\_\_ ZIP: \_\_\_\_\_

COUNTRY: \_\_\_\_\_

PHONE NO.: ( ) \_\_\_\_\_

ORDER NO.	TITLE	QTY.	PRICE	TOTAL
<input type="text"/>	_____	_____ x	_____ =	_____
<input type="text"/>	_____	_____ x	_____ =	_____
<input type="text"/>	_____	_____ x	_____ =	_____
<input type="text"/>	_____	_____ x	_____ =	_____
<input type="text"/>	_____	_____ x	_____ =	_____
<input type="text"/>	_____	_____ x	_____ =	_____
<input type="text"/>	_____	_____ x	_____ =	_____
<input type="text"/>	_____	_____ x	_____ =	_____
<input type="text"/>	_____	_____ x	_____ =	_____
<input type="text"/>	_____	_____ x	_____ =	_____

Subtotal \_\_\_\_\_

Must Add Your  
Local Sales Tax \_\_\_\_\_

Include postage:  
Must add 15% of Subtotal to cover U.S.  
and Canada postage. (20% all other.)

→ Postage \_\_\_\_\_

Total \_\_\_\_\_

Pay by check, money order, or include company purchase order with this form (\$200 minimum). We also accept VISA, MasterCard or American Express. Make payment to Intel Literature Sales. Allow 2-3 weeks for delivery.

☐ VISA ☐ MasterCard ☐ American Express Expiration Date \_\_\_\_\_

Account No. \_\_\_\_\_

Signature \_\_\_\_\_

**Mail To:** Intel Literature Sales  
P.O. Box 7641  
Mt. Prospect, IL 60056-7641

**International Customers** outside the U.S. and Canada should use the International order form on the next page or contact their local Sales Office or Distributor.

**For phone orders in the U.S. and Canada**  
**Call Toll Free: (800) 548-4725**

Prices good until 12/31/93.  
Source HB

CG/LOF1/103092



## INTERNATIONAL LITERATURE ORDER FORM

NAME: \_\_\_\_\_

COMPANY: \_\_\_\_\_

ADDRESS: \_\_\_\_\_

CITY: \_\_\_\_\_ STATE: \_\_\_\_\_ ZIP: \_\_\_\_\_

COUNTRY: \_\_\_\_\_

PHONE NO.: ( ) \_\_\_\_\_

ORDER NO.	TITLE	QTY.	PRICE	TOTAL
<input type="text"/>	_____	_____	_____	_____
<input type="text"/>	_____	_____	_____	_____
<input type="text"/>	_____	_____	_____	_____
<input type="text"/>	_____	_____	_____	_____
<input type="text"/>	_____	_____	_____	_____
<input type="text"/>	_____	_____	_____	_____
<input type="text"/>	_____	_____	_____	_____
<input type="text"/>	_____	_____	_____	_____
<input type="text"/>	_____	_____	_____	_____
<input type="text"/>	_____	_____	_____	_____

Subtotal \_\_\_\_\_

Must Add Your Local Sales Tax \_\_\_\_\_

Total \_\_\_\_\_

### PAYMENT

Cheques should be made payable to your **local** Intel Sales Office (see inside back cover).

Other forms of payment may be available in your country. Please contact the Literature Coordinator at your **local** Intel Sales Office for details.

The completed form should be marked to the attention of the LITERATURE COORDINATOR and returned to your **local** Intel Sales Office.

International Customers outside the U.S. and Canada should use the International Order Form on the next page or contact their local Intel Sales Office for details.

Mail To: Intel Literature Sales  
P.O. Box 7841  
Mt. Prospect, IL 60058-7841

For phone orders in the U.S. and Canada  
Call Toll Free: (800) 548-4725





# PROGRAMMABLE LOGIC

1993

MAFNET™	1980™	Above™
Matched™	1980™	ActionMedia™
MOSE™	1980™	GLIBUS™
Media Mail™	1980™	Code Builder™
NetPort™	1980™	Desktop™
NetPort™	1980™	Digital Studio™
OpenNET™	1980™	QV™
OverDrive™	1980™	Smartness™
Paragon™	1980™	ETOX™
Perform™	1980™	EXA™
ProPower™	1980™	Graphics and Go™
RadCAD™	1980™	IMAX™
READY-LAN™	1980™	ISBX™
Reference Point™	1980™	ISBX™
RMX-80™	1980™	ISBX™
Server™	1980™	ISBX™
SealAX™	1980™	ISBX™
Shash 386™	1980™	ISBX™
Storage Broker™	1980™	ISBX™
SubC™	1980™	ISBX™
The Computer Inside™	1980™	ISBX™
TowerExpress™	1980™	ISBX™
Ultra Edge™	1980™	ISBX™
WYNYWYF™	1980™	ISBX™

MDS is an ordering code only and is not used as a product name or trademark. MDS is a registered trademark of Microware Data Sciences Corporation.

CHMOS and HMOS are patented processes of Intel Corp.

Intel Corporation and Intel's FASTPATH are not affiliated with Inteltek, a division of Excelan, Inc. or its FASTPATH technologies or products.

Additional copies of this manual or other Intel literature may be obtained from:

Intel Corporation  
Literature Sales  
P.O. Box 7641  
Mt. Prospect, IL 60056-7641

© 1993 Intel Corporation

00-11-000

Intel Corporation makes no warranty for the use of its products and assumes no responsibility for any errors which may appear in this document nor does it make a commitment to update the information contained herein.

Intel retains the right to make changes to these specifications at any time, without notice.

Contact your local sales office to obtain the latest specifications before placing your order.

The following are trademarks of Intel Corporation and may only be used to identify Intel products:

376™	i750®	MAPNET™
Above™	i860™	Matched™
ActionMedia®	i960®	MCS®
BITBUS™	Intel287™	Media Mail™
Code Builder™	Intel386™	NetPort™
DeskWare™	Intel387™	NetSentry™
Digital Studio™	Intel486™	OpenNET™
DVI®	Intel487™	OverDrive™
EtherExpress™	Intel®	Paragon™
ETOX™	intel inside®	Pentium™
ExCA™	Intellec®	ProSolver™
Exchange and Go™	iPSC®	RapidCAD™
FaxBACK™	iRMX®	READY-LAN™
FlashFile™	iSBC®	Reference Point®
Grand Challenge™	iSBX™	RMX/80™
j®	iWARP™	RxServer™
ICE™	LANDesk™	SatisFAXtion®
iLBX™	LANPrint®	SnapIn 386™
Inboard™	LANProtect™	Storage Broker™
i287™	LANSelect®	SugarCube™
i386™	LANShell®	The Computer Inside™
i387™	LANsight™	TokenExpress™
i486™	LANSpace®	Visual Edge™
i487™	LANSpool®	WYPIWYF®

MDS is an ordering code only and is not used as a product name or trademark. MDS is a registered trademark of Mohawk Data Sciences Corporation.

CHMOS and HMOS are patented processes of Intel Corp.

Intel Corporation and Intel's FASTPATH are not affiliated with Kinetics, a division of Excelan, Inc. or its FASTPATH trademark or products.

Additional copies of this manual or other Intel literature may be obtained from:

Intel Corporation  
Literature Sales  
P.O. Box 7641  
Mt. Prospect, IL 60056-7641





## PLD CUSTOMER SUPPORT

### NEW TECHNICAL LITERATURE

By the time you receive this handbook, new technical information will be available on Intel PLDs. It may include advanced information on new devices or important new applications information for devices described in the handbook. If you want the newest information on Intel PLDs, please call the toll free Literature number listed at the front of this handbook and ask for Literature Item IG69.

### EPLD HOTLINE

The Intel EPLD Technical Hotline is manned by applications personnel during normal business hours. You can leave a message during off hours or when applications personnel are already handling calls. The number (U.S. and Canada) is 1-800-323-EPLD (1-800-323-3753).

### BBS

Intel has a Bulletin Board System for registered PLDshellPlus customers to electronically transfer information. A registered user with a modem can log onto the system. The current number is (916) 985-2308. If your communication software supports file transfers, you can receive utilities, software updates, and the latest information on EPLDs via the Bulletin Board.

## DATA SHEET DESIGNATIONS

Intel uses various data sheet markings to designate each phase of the document as it relates to the product. The marking appears in the upper, right-hand corner of the data sheet. The following is the definition of these markings:

Data Sheet Marking	Description
Product Preview	Contains information on products in the design phase of development. Do not finalize a design with this information. Revised information will be published when the product becomes available.
Advanced Information	Contains information on products being sampled or in the initial production phase of development.*
Preliminary	Contains preliminary information on new products in production.*
No Marking	Contains information on products in full production.*

\*Specifications within these data sheets are subject to change without notice. Verify with your local Intel sales office that you have the latest data sheet before finalizing a design.



Page	Device
5-3	FX780
5-38	FX740
5-38	FX7180
5-31	PLD810/88C080
5-30	PLD810/88C080
5-32	PLD82V10
5-35	88C52V10
5-30	88C520/88C52A-100, -80, and -68
5-117	88C550/88C55A-100, -80, and -68
5-128	88C550/88C55A-100, -80, and -68
5-143	88C550/88C55A-100, -80, and -68
5-181	88C550/88C55A-100, -80, and -68
5-174	88C550/88C55A-100, -80, and -68
5-192	88C550/88C55A-100, -80, and -68
5-208	88C550/88C55A-100, -80, and -68

## Overview

## Device Data Sheets

## Applications Information

## Development Support Tools

## Appendix

1

2

3

4

5

## Intel PLDs

Device	.....	Page
iFX780	.....	2-3
iFX740	.....	2-25
iFX7160	.....	2-26
iPLD610/85C060	.....	2-31
iPLD910/85C090	.....	2-50
iPLD22V10	.....	2-72
85C22V10	.....	2-82
85C220/85C224-100, -80, and -66	.....	2-93
85C220/85C224-7 and -10	.....	2-111
5AC312	.....	2-125
5AC324	.....	2-143
5C032	.....	2-161
5C060	.....	2-174
5C090	.....	2-192
5C180	.....	2-209

Order Number: 296832-002



# Table of Contents

Alphanumeric Index .....	xiii
<b>CHAPTER 1</b>	
<b>Overview</b>	
Overview .....	1-1
<b>CHAPTER 2</b>	
PLDs at a Glance .....	2-1
<b>Device Data Sheets</b>	
<b>FLEXlogic FPGA Family</b>	
iFX780 Advanced Information Datasheet .....	2-3
iFX740 Product Preview Datasheet .....	2-25
iFX7160 Product Preview Datasheet .....	2-26
FPGA Tutorial .....	2-27
<b>High-Performance Microcomputer Programmable Logic Devices</b>	
iPLD610/85C060 Fast 16-Macrocell CMOS PLD .....	2-31
iPLD910/85C090 Fast 24-Macrocell CMOS PLD .....	2-50
iPLD22V10 Fast 10-Macrocell CMOS PLD .....	2-72
85C22V10 Fast 10-Macrocell CHMOS microPLD .....	2-82
85C220/85C224 Family Product Selector Guide .....	2-92
—16V8/20V8-Type Architecture	
85C220/85C224-100, -80 and -66 Register Optimized Timing Fast 1-Micron	
CHMOS 8-Macrocell microPLDs .....	2-93
—100 MHz Meets Advanced Intel Microprocessor Timing Requirements	
—80 MHz Meets Intel 486/386-Based System Timing; Zero Power	
—66 MHz Meets High-Performance Microprocessor and Hi-Speed PLD Timing	
85C220/85C224-7 and -10 Combinatorial Optimized Timing Fast 1-Micron CHMOS	
8-Macrocell microPLDs .....	2-111
—7 ns High-Performance and Low Power Upgrade for -7 PALs/GALs	
<b>Advanced Architecture EPLDs</b>	
5AC312 1-Micron CHMOS 12-Macrocell EPLD .....	2-125
5AC324 1-Micron CHMOS 24-Macrocell EPLD .....	2-143
<b>Standard Architecture EPLDs</b>	
5C032 8-Macrocell CHMOS EPLD .....	2-161
5C060 16-Macrocell CHMOS EPLD .....	2-174
5C090 24-Macrocell CHMOS EPLD .....	2-192
5C180 48-Macrocell CHMOS EPLD .....	2-209
<b>CHAPTER 3</b>	
<b>Applications Information</b>	
<b>High-Performance Microcomputer Programmable Logic Devices</b>	
APPLICATION NOTES	
AP-338 85C220/85C224 Design Guide .....	3-1
AP-339 iPLD610 Design Guide .....	3-18
AP-371 iPLD22V10/85C22V10 Design Guide .....	3-68
<b>Advanced Architecture EPLDs</b>	
APPLICATION NOTES	
AP-317 Implementing a PS/2 POS Using the 5AC312 EPLD .....	3-88
AP-319 Designing with the 5AC312/5AC324 EPLDs .....	3-101
ARTICLE REPRINTS	
AR-476 A Micro Channel Slave-Adapter Link .....	3-111

## Table of Contents (Continued)

AR-489 Simplify a RISC-Embedded Controller Interface Using a PLD .....	3-112
<b>TECHNICAL PAPERS</b>	
Programmable AND/Allocatable OR Based EPLD Addresses the Needs of Complex Combinational and Sequential Designs .....	3-119
Advanced Architecture PLDs Solve Common State Machine Problems .....	3-126
A 15 ns 2500 Gate Highly Flexible CHMOS EPLD .....	3-130
<b>Standard Architecture EPLDs</b>	
<b>APPLICATION BRIEFS</b>	
AB-8 Implementing Cascaded Logic in EPLDs .....	3-134
AB-11 16-Bit Binary Counter Implementation Using the 5C060 EPLD .....	3-138
AB-12 Designing a Mailbox Memory for Two 80C31 Microcontrollers Using EPLDs ..	3-149
AB-16 Atypical Latch/Register Construction in EPLDs .....	3-158
<b>APPLICATION NOTES</b>	
AP-272 The 5C060 Unification of a CHMOS System .....	3-166
AP-276 Implementing a CMOS Bus Arbiter/Controller in the 5C060 EPLD .....	3-178
AP-321 Fitting the 5C180 .....	3-189
<b>General</b>	
AP-307 EPLDs, PLAs and TTL Comparing the "Hidden Costs" in Production .....	3-203
AP-336 Metastability Characteristics of Intel EPLDs .....	3-226
RR-64 PLD Quality and Reliability Data Summary .....	3-234
<b>CHAPTER 4</b>	
<b>Development Support Tools</b>	
PLD Support at a Glance .....	4-1
<b>DATA SHEETS</b>	
Development Tools Overview .....	4-3
PLDshell Plus Design Software .....	4-17
iUP-PC Universal Programmer for the Personal Computer .....	4-20
iUP-200A/iUP-201A Universal PROM Programmers .....	4-26
<b>ARTICLE REPRINT</b>	
AR-703 Personalizing PLD Design Work .....	4-32
<b>CHAPTER 5</b>	
<b>Appendix</b>	
PLD Programming Support .....	5-1
PLD Programming Vendors .....	5-3
PLD Software Support .....	5-5
PLD Software Vendors .....	5-6
PAL/GAL to Intel PLD Replacement .....	5-7
Intel PLD Feature Comparison .....	5-8
Industrial Temperature Devices .....	5-9
Military Devices .....	5-9
Tape and Reel Packaging .....	5-10
Pre-Programmed Devices .....	5-10
Register Preload .....	5-10
Ordering Information .....	5-11
EPLD Customer Support .....	5-12



## Alphanumeric Index

5AC312 1-Micron CHMOS 12-Macrocell EPLD .....	2-125
5AC324 1-Micron CHMOS 24-Macrocell EPLD .....	2-143
5C032 8-Macrocell CHMOS EPLD .....	2-161
5C060 16-Macrocell CHMOS EPLD .....	2-174
5C090 24-Macrocell CHMOS EPLD .....	2-192
5C180 48-Macrocell CHMOS EPLD .....	2-209
85C220/85C224 Family Product Selector Guide .....	2-92
85C220/85C224-100, -80 and -66 Register Optimized Timing Fast 1-Micron CHMOS 8-Macrocell microPLDs .....	2-93
85C220/85C224-7 and -10 Combinatorial Optimized Timing Fast 1-Micron CHMOS 8-Macrocell microPLDs .....	2-111
85C22V10 Fast 10-Macrocell CHMOS microPLD .....	2-82
A 15 ns 2500 Gate Highly Flexible CHMOS EPLD .....	3-130
AB-11 16-Bit Binary Counter Implementation Using the 5C060 EPLD .....	3-138
AB-12 Designing a Mailbox Memory for Two 80C31 Microcontrollers Using EPLDs .....	3-149
AB-16 Atypical Latch/Register Construction in EPLDs .....	3-158
AB-8 Implementing Cascaded Logic in EPLDs .....	3-134
Advanced Architecture PLDs Solve Common State Machine Problems .....	3-126
AP-272 The 5C060 Unification of a CHMOS System .....	3-166
AP-276 Implementing a CMOS Bus Arbiter/Controller in the 5C060 EPLD .....	3-178
AP-307 EPLDs, PLAs and TTL Comparing the "Hidden Costs" in Production .....	3-203
AP-317 Implementing a PS/2 POS Using the 5AC312 EPLD .....	3-88
AP-319 Designing with the 5AC312/5AC324 EPLDs .....	3-101
AP-321 Fitting the 5C180 .....	3-189
AP-336 Metastability Characteristics of Intel EPLDs .....	3-226
AP-338 85C220/85C224 Design Guide .....	3-1
AP-339 iPLD610 Design Guide .....	3-18
AP-371 iPLD22V10/85C22V10 Design Guide .....	3-68
AR-476 A Micro Channel Slave-Adapter Link .....	3-111
AR-489 Simplify a RISC-Embedded Controller Interface Using a PLD .....	3-112
AR-703 Personalizing PLD Design Work .....	4-32
Development Tools Overview .....	4-3
FPGA Tutorial .....	2-27
iFX7160 Product Preview Datasheet .....	2-26
iFX740 Product Preview Datasheet .....	2-25
iFX780 Advanced Information Datasheet .....	2-3
iPLD22V10 Fast 10-Macrocell CMOS PLD .....	2-72
iPLD610/85C060 Fast 16-Macrocell CMOS PLD .....	2-31
iPLD910/85C090 Fast 24-Macrocell CMOS PLD .....	2-50
iUP-200A/iUP-201A Universal PROM Programmers .....	4-26
iUP-PC Universal Programmer for the Personal Computer .....	4-20
Overview .....	1-1
PLD Programming Support .....	5-1
PLD Support at a Glance .....	4-1
PLDs at a Glance .....	2-1
PLDshell Plus Design Software .....	4-17
Programmable AND/Allocatable OR Based EPLD Addresses the Needs of Complex Combinational and Sequential Designs .....	3-119
RR-64 PLD Quality and Reliability Data Summary .....	3-234







Overview

# Programmable Logic Overview

PAGE	CONTENTS	PAGE	CONTENTS
1-1	LOGIC REFRESHER COURSE	1-1	INTRODUCTION
1-2	Boolean Algebra	1-2	WHY USER DEFINED LOGIC?
1-3	Karnaugh Maps	1-3	USER-DEFINED LOGIC IMPLEMENTATION
1-4	AUTOMATIC STANDBY MODE (TURBO BIT)	1-4	ALTERNATIVES
1-5	Turbo Off (Low Power)	1-5	PROGRAMMABLE LOGIC
1-6	Turbo On (Faster Speed)	1-6	IMPLEMENTATION OF BIPOLAR FUSE
1-7	PAI/GAL TO PLD REPLACEMENT	1-7	ERASABLE PROGRAMMABLE LOGIC DEVICES
1-8	85C250	1-8	CMOS TECHNOLOGY IN EPLDs
1-9	85C254	1-9	CMOS DESIGN GUIDELINES
1-10	85C250/PLD810	1-10	Electrostatic Discharge
1-11	85C254/PLD2510	1-11	PCB Layout
1-12	QUALITY/RELIABILITY	1-12	Decoupling
1-13	PACKAGING	1-13	Unused Inputs
1-14	SOFTWARE AND PROGRAMMING SUPPORT	1-14	BOOLEAN MINIMIZATION FOR PLA ARCHITECTURES
1-15		1-15	References

## Programmable Logic Overview

1

October 1992

## Programmable Logic Overview

CONTENTS	PAGE	CONTENTS	PAGE
INTRODUCTION .....	1-3	LOGIC REFRESHER COURSE .....	1-8
WHY USER DEFINED LOGIC? .....	1-3	Boolean Algebra .....	1-9
USER-DEFINED IC— IMPLEMENTATION ALTERNATIVES .....	1-3	Karnaugh Maps .....	1-9
PROGRAMMABLE LOGIC .....	1-4	AUTOMATIC STANDBY MODE (TURBO BIT) .....	1-11
LIMITATIONS OF BIPOLAR FUSE TECHNOLOGY FOR PLDs .....	1-5	Turbo Off (Low Power) .....	1-11
ERASABLE PROGRAMMABLE LOGIC DEVICES .....	1-5	Turbo On (Faster Speed) .....	1-11
CHMOS TECHNOLOGY IN EPLDs .....	1-7	PAL/GAL TO PLD REPLACEMENT ....	1-12
CHMOS DESIGN GUIDELINES .....	1-7	85C220 .....	1-12
Electrostatic Discharge .....	1-7	85C224 .....	1-12
PCB Layout .....	1-8	85C060/PLD610 .....	1-12
Decoupling .....	1-8	85C22V10/PLD22V10 .....	1-12
Unused Inputs .....	1-8	QUALITY/RELIABILITY .....	1-13
BOOLEAN MINIMIZATION TECHNIQUE FOR PLA ARCHITECTURES .....	1-8	PACKAGING .....	1-13
References .....	1-8	SOFTWARE AND PROGRAMMING SUPPORT .....	1-13
		ORDERING INFORMATION .....	1-14



In 1989, Intel introduced a new line of high-speed Microcomputer Programmable Logic Devices ( $\mu$ PLDs) aimed at augmenting high-performance microcomputer environments. This new family of devices provides high-speed support logic for the fast microcomputer systems of today. At the same time, our advanced CMOS process allows our PLDs to significantly reduce power consumption and system heat dissipation, problems that plagued the previous generation of bipolar PLDs.

Major benefits of Intel's  $\mu$ PLD family are:

Industry's Highest Speed PLDs	<ul style="list-style-type: none"> <li>• High Speed</li> <li>— 100 MHz + Registered</li> <li>— &lt; 7 ns tpd</li> </ul>
Specially Designed Register and Output Circuits	<ul style="list-style-type: none"> <li>• Lowest Noise CMOS for Simplified High-Speed System Design</li> <li>• Best Solution for Metastable Conditions</li> </ul>
Superset of Popular CMOS and Bipolar PLDs	<ul style="list-style-type: none"> <li>• Compatible with Existing Design Methods</li> <li>• Fewer Manufacturing Line Items</li> </ul>
Built on Standard CHMOS EPROM Technology	<ul style="list-style-type: none"> <li>• High Reliability/Quality</li> <li>— Lower Failure Rates</li> <li>— 100% Programming Yields</li> </ul>

Intel's high-performance  $\mu$ PLDs, along with the existing EPLD family of devices, can be classified as "User-Defined Logic" circuits. User-defined logic circuits allow system designers to tailor building block solutions to their individual systems requirements. This customization provides the needed performance, reliability, and space reduction as well as design security.

This document discusses the reasons for the trend to user-defined logic devices, briefly describes some implementation alternatives, and provides detail on a solution that can be implemented completely by the user, i.e., the programmable logic device. Details on Intel's PLD product line, including terminology, nomenclature, architectural features, and developmental tools, are also described in this document.

## WHY USER DEFINED LOGIC?

System designers prefer user customized ICs for the following reasons:

space, resulting in smaller system physical dimensions.

**b. LOWER SYSTEM COSTS:** When custom LSI or VLSI components are used instead of standard SSI and MSI logic elements, there is a considerable saving in component cost per system, assembly and manufacturing cost, printed circuit board area and board costs and inventory costs.

**c. HIGHER PERFORMANCE:** Reduced number of ICs contributes to faster system speeds as well as lower power consumption.

**d. HIGHER RELIABILITY:** Since probability of failure is directly related to the number of ICs in the system, a system composed of customized LSI & VLSI chips is statistically much more reliable than the identical system made up of SSI/MSI devices.

**e. DESIGN SECURITY:** Systems designed with standard components can be replicated relatively easily whereas systems that contain user customized ICs cannot be copied because "reverse engineering" of the customized components is extremely difficult. Thus, use of customized ICs allows for the protection of proprietary designs.

**f. INCREASED FLEXIBILITY:** Customized components allow for the tailoring of systems to the end user's specific needs relatively easily. This also allows for upgradability and obsolescence protection.

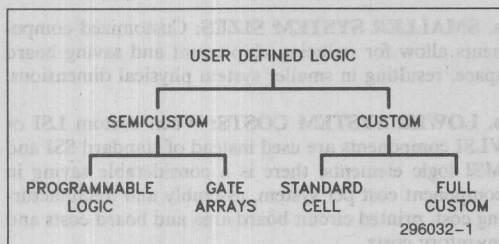
## USER DEFINED IC—IMPLEMENTATION ALTERNATIVES

Currently, the choices available to the system designer for customization of ICs (see Figure 1) are as follows:

- (1) user programmable ICs—programmable logic devices, including Field Programmable Gate Arrays (FPGAs)
- (2) mask programmable ICs—gate arrays
- (3) standard cell based ICs
- (4) full custom ICs

Alternatives (1) & (2) are usually called 'Semicustom' because in these methods only a few (less than three) of the mask layers involved in the manufacture of the IC, are customized to the users' specifications. The later two alternatives (3) & (4), involve customization of all mask layers required to manufacture the ICs to the users' specifications and are therefore called 'Custom'.

1



**Figure 1. User-Defined Logic Implementation Choices**

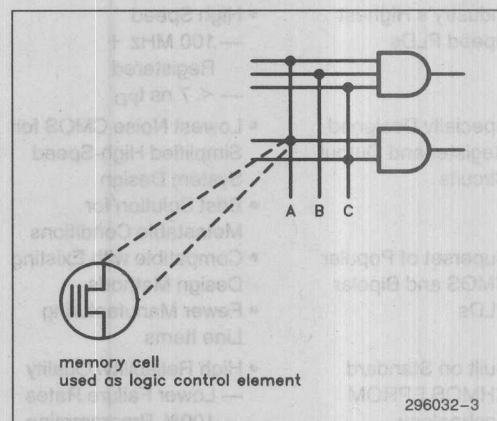
## PROGRAMMABLE LOGIC

Most user Programmable Logic Devices (PLD) are internally structured as variations of the PLA (programmable logic array) architecture, that is composed of an array of 'AND' gates connected to an array of 'OR' gates (see Figure 2). Programmable logic devices make use of the fact that any logic equation can be converted to an equivalent 'Sum-of-Products' form and can thus be implemented in the 'AND' and 'OR' architecture. This basic PLA structure has been augmented in most PLDs with input and output blocks containing registers, latches and feedback options, that let the user implement sequential logic functions in addition to combinational logic.

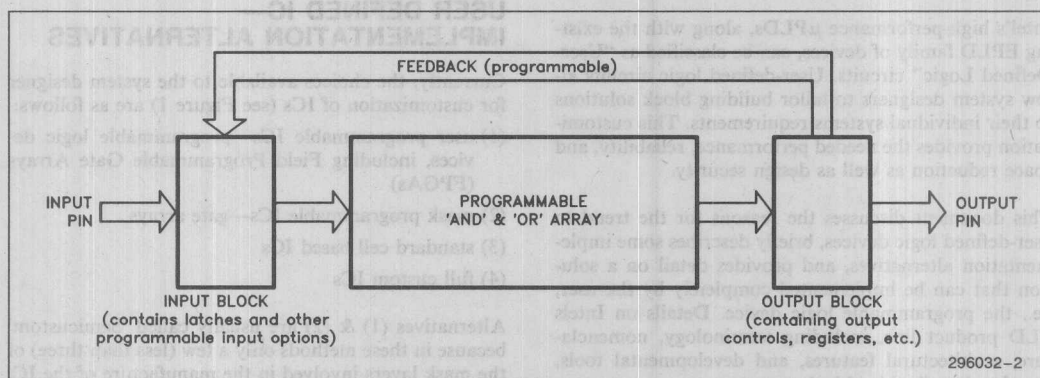
The number and locations of the programmable connections between the 'AND' and 'OR' matrices as well as the input and output blocks are predetermined by the architecture of the PLD. The user, depending on

his logic requirements, determines which of these connections he would like to remain open and which he would like to close, through the programming of the PLD. Programmability of these connections is achieved using various memory technologies such as fuses, EPROM cells, EEPROM cells or Static RAM cells (see Figure 3).

User programmability allows for instant customization, very similar to user programmable memories such as PROMs or EPROMs. The user can purchase a PLD off-the-shelf, use a development system running on a personal computer and, in a matter of a few hours, have customized silicon in his hands. Figure 4 compares user-defined logic alternatives.



**Figure 3. Programmable Connections**



**Figure 2. General Architecture of a PLD**

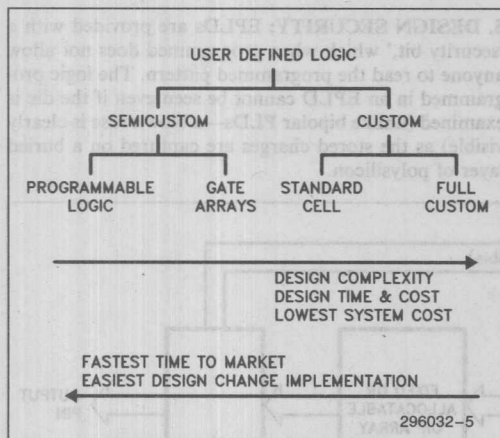


Figure 4. User-Defined Logic Alternatives Compared

## LIMITATIONS OF BIPOLAR FUSE TECHNOLOGY FOR PROGRAMMABLE LOGIC DEVICES

Until 1985, all PLDs were built using Bipolar fuse technology. The bipolar fuse based devices, although offering the users the benefits of quick time to market and low development costs, had several inherent limitations.

**a. HIGH POWER CONSUMPTION:** Bipolar processes by nature are power hungry and as a consequence also make for very hot systems, often requiring cooling aids such as heat sinks and fans. They also cannot operate at lower voltages (2–3V) and have a lower level of noise immunity than MOS devices.

**b. LOWER INTEGRATION:** A fuse takes up a large amount of silicon area; this fact in conjunction with the large power requirements makes for smaller levels of integration.

**c. ONE-TIME PROGRAMMABILITY:** Bipolar fuses can only be blown once and cannot be reprogrammed. This does not allow for easy prototyping and could result in significant losses when preprogrammed parts are inventoried and design changes occur.

**d. TESTABILITY:** Since fuses can only be blown once, bipolar PLDs can only be destructively tested. Thus, testing is usually done by sampling or through addi-

tional testing elements incorporated in the chips, which can be blown to examine electrical characteristics. However, such testing methods never allow for 100% testability of all parts shipped. Thus, most users of bipolar programmable logic devices resort to extensive post-programming testing, specific to their applications.

## ERASABLE PROGRAMMABLE LOGIC DEVICES

Erased programmable logic devices (EPLD) result from the matching of CHMOS EPROM technology with the architectures of programmable logic devices. EPLDs use EPROM cells as logic control elements and therefore, when housed in windowed ceramic packages, can be erased with UV light and reprogrammed. Figure 5 shows the architecture of Intel EPLDs.

Other than the obvious benefit of reprogrammability, EPLDs offer several very significant benefits over bipolar PLDs. These are:

**1. LOW POWER CONSUMPTION:** Due to the CMOS technology, these products consume an order of magnitude less power than the equivalent bipolar devices. This allows for the design of complete CMOS systems, that can operate at lower voltages (less than 5V). Also, this makes for cooler systems that do not require cooling systems like fans.

**2. GREATER LOGIC DENSITY:** EPROM cells are an order of magnitude smaller than the smallest fuses. This means that the same function can be accommodated in significantly smaller die area, or that greater amounts of logic can now be incorporated on a single chip. Thus higher integration programmable logic devices result with the use of EPROM elements.

**3. TESTABILITY:** Since the EPROM cells are erasable, the entire EPROM array of the EPLD can be 100% factory tested. Thus, before the part is shipped to the customers, it can be completely tested by the programming and erasure of all the EPROM logic control bits. This testing is therefore independent of any application, in contrast to the bipolar PLDs that need application specific testing.

**4. ARCHITECTURAL ENHANCEMENTS:** The inherent testability of the EPROM elements allows for



significant architectural improvements over bipolar PLDs. New features, such as buried registers, programmable registers, programmable clock control, etc., can now be incorporated because of this testability. These new features allow for greatly increased utilization of the EPLDs and use of these devices in newer applications.

**5. DESIGN SECURITY:** EPLDs are provided with a 'security bit,' which when programmed does not allow anyone to read the programmed pattern. The logic programmed in an EPLD cannot be seen even if the die is examined (unlike bipolar PLDs—a blown fuse is clearly visible) as the stored charges are captured on a buried layer of polysilicon.

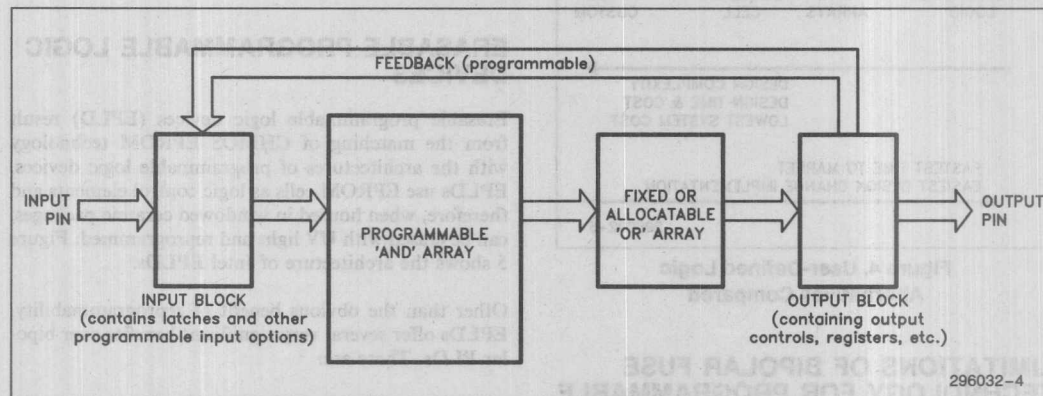


Figure 5. Architecture of Intel EPLDs

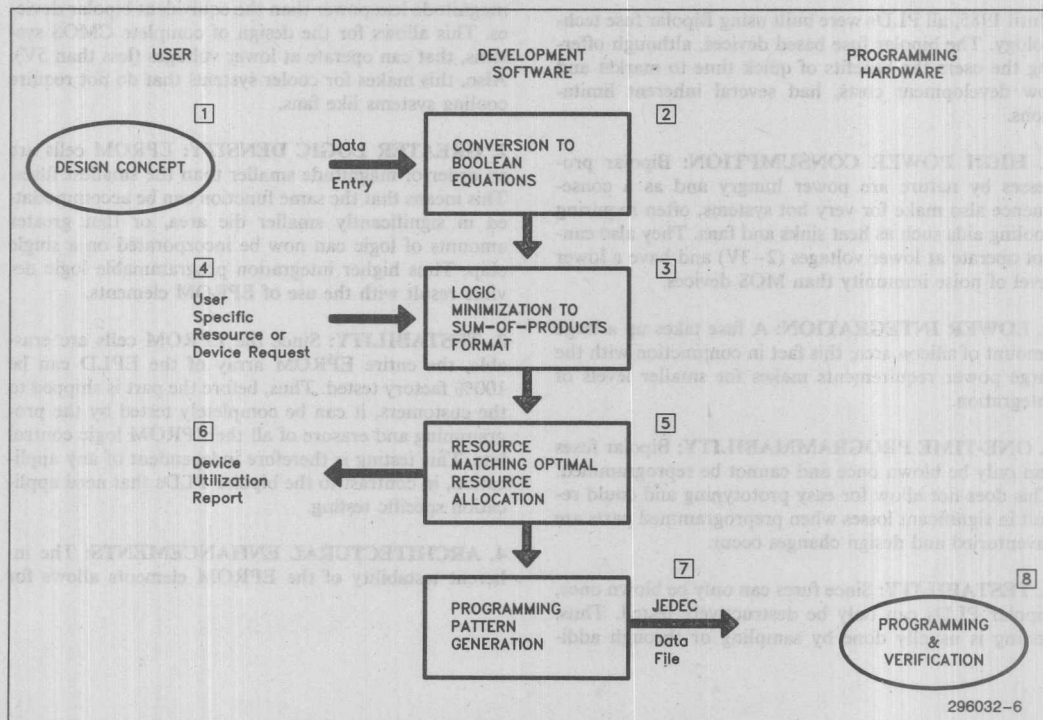


Figure 6. The PLD Design Process

The steps in a generalized design process of programmable logic is shown in Figure 6 and described in the following paragraphs.

**STEP 1:** The user decides on the logic he wants implemented in the PLD and enters the design into the PC or workstation. This **Design Entry** may be done by the following methods: (i)**SCHEMATIC CAPTURE**—A 'Mouse' or some other graphics input device is used to input schematics of the logic, (ii)**NET LIST ENTRY**—If the user has a hand drawn schematic he can enter the design into the computer by describing the symbols and interconnections in words using a standardized format called a net list (without using a graphics input device), (iii)**STATE EQUATION/DIAGRAM ENTRY**—Entry of a sequential design involving states and transitions between states. In the state diagram method circles represent states and the arrows interconnecting them represent the transitions. Equations or a state table can also be used to define a state machine, and (iv)**BOOLEAN EQUATIONS**—this is the most common design entry method. The logic is described in boolean algebraic equations.

**STEP 2:** The software converts all design entry data into boolean equations.

**STEP 3:** The boolean equations entered are converted to the sum of products format after logic reduction (minimization of the logic through heuristic algorithms).

**STEP 4:** The user has the ability to choose the PLD he would like the design implemented on. He can enter device choice and/or he can also enter in specific choices on the device as regards pinout he would like etc . . .

**STEP 5:** The software optimizes the logic equations to fit into the device using the minimum amount of resources (resources are input pins, output pins, registers and product terms and macrocells). This step is where the user requirements as regards required pins are taken into account. The user requests are viewed as constraints during the optimization process.

**STEP 6:** The software, at the end of the resource optimization/allocation, produces a report detailing the resources used up in fitting the design on the PLD. This report allows the user to incrementally stuff in logic by going back to Step 1 from this stage. Also, if the design overflowed the PLD, i.e., did not fit in the user chosen device, the software lists out the resources needed to complete the fit. The requirements such as four more inputs, one register more and one more output (are needed to complete the design) gives the user data in choosing a bigger PLD or in partitioning the initial design to fit into two devices.

**STEP 7:** The next step is to generate the appropriate programming pattern for the PLD. This is a standard

"JEDEC" format interface and allows the output of the design software to be compatible with any piece of PROM programming hardware.

**STEP 8:** PROM programmer is used to program the pattern stored in the JEDEC file onto the PLD. Also, at this stage fuse programmed PLDs (bipolar) are functionally tested using test vectors included in the JEDEC file information.

## CHMOS TECHNOLOGY IN EPLDs

EPLDs are manufactured with Intel's proprietary CHMOS (Complementary High-Performance MOS) technology. The backbone of the process is the integration of both a P and an N channel MOS transistor on the same substrate. In addition, EPLD's programmable architecture makes use of Intel's proven EPROM cell for programmable array interconnections as well as macrocell configuration bits. These cells are programmed electrically and erased with ultraviolet light. For details on Intel's CHMOS technology and EPROM cells technology, refer to the *Components Quality/Reliability Handbook*, Order Number 210997.

## CHMOS DESIGN GUIDELINES

Designing with Intel EPLDs is relatively straightforward if the following guidelines are observed:

- Minimize the occurrence of ESD (electro-static discharge) when storing or handling EPLDs.
- Observe good design rules in printed circuit board layout.
- Provide adequate decoupling capacitance at both the device and the board level.
- Connect all unused inputs to  $V_{CC}$  or GND (CHMOS inputs should not be left floating).

## Electrostatic Discharge

The two most common sources of electrostatic discharge are the human body and a charged environment.

A charged human body that touches a device lead discharges electricity into the device. Electrostatic discharge from people handling devices has long been recognized by manufacturers and users of all MOS products. Human body static electricity can be controlled by using ground straps and anti-static spray on carpeted floors. CHMOS devices should also be stored and carried in conductive tubes or anti-static foam to minimize exposure to ESD from people.

Discharge also occurs when an integrated circuit is charged to one potential and then contacts a conductor at another potential. This type of ESD can be reduced

by grounding all work surfaces, grounding all handling equipment, removing static generators such as paper from the work area, and erasing EPLDs in metal tubes, metal trays, or conductive foam.

## PCB Layout

The best PCB performance is obtained when close attention is paid to  $V_{CC}$ , GND, and signal traces.  $V_{CC}$  and GND should be gridded to minimize inductive reactance and to approximate a trace layer. Clocks should be laid out to minimize crosstalk. Ensure adequate power supply and ground pins on the board connector.

## Decoupling

Decouple each EPLD with a high-frequency ceramic capacitor in the range of 0.01  $\mu\text{F}$  to 0.2  $\mu\text{F}$ , depending on board frequency and current consumption. For most applications, a 0.1  $\mu\text{F}$  capacitor will suffice. The following equation produces the exact value:

$$C = \frac{\Delta I_{CC}}{\Delta V / \Delta T}$$

where  $C$  = capacitor value

$\Delta I_{CC}$  = maximum switched current

$\Delta V$  = switching level

$\Delta T$  = switching time

For boards that contain mixed logic (EPLDs and TTL), observe both EPLD and TTL decoupling practices.

## Unused Inputs

To minimize noise receptivity and power consumption, all unused inputs to EPLDs should be connected to  $V_{CC}$  or GND. By default, PLDshell Plus software assigns unused inputs to GND. These pins shown on the pinout representation of the PLDshell Plus report file, should be connected to ground on the PCB. Pins listed as RESERVED on the report file must be left floating. Pins marked N.C. have no internal device connections and can also be left floating.

## BOOLEAN MINIMIZATION TECHNIQUES FOR SOP ARCHITECTURES

Minimization plays an important role in logic design. Methods for minimization can be grouped into two classes. Class 1 includes manual methods for minimization, such as Boolean reduction or Karnaugh mapping. Class 2 is computer-assisted minimization.

Tabular methods like Karnaugh maps are efficient up to a certain point. Past that point, however, computer-assisted minimization plays a crucial part in efficient design. Even at the computer-assisted stage, the choice of minimizer software has an impact on time and the confidence level of the reduced equation (i.e., is it in the smallest possible form).

PLDshell Plus software includes a minimizer that uses the ESPRESSO algorithms. ESPRESSO was developed by U.C. Berkeley. The primary advantage of the ESPRESSO minimizer becomes apparent when designing large finite state machines or complex, product-term intensive logic designs. In these cases, ESPRESSO arrives at the minimize solution sooner, and frequently reduces the logic to a smaller number of product terms.

For more information on ESPRESSO, refer to *Logic Minimization Algorithms for VLSI Synthesis*, Brayton, Hachtel, McMullen, and Sangiovanni-Vincentelli, Kluwer Academic Publishers.

## References

- [BRO 81] D.W. Brown, "A State-Machine Synthesizer—SMS", Proc. 18th Design Automation Conference, pp. 301–304. Nashville, June 1981.
- [HON 74] S. J. Hong, R. G. Cain and D. L. Ostapko, "MINI": A heuristic approach to logic minimization." *IBM Journal of Research and Development*, Vol. 18, pp. 443–458, September 1974.

ABEL™ is a trademark of Data I/O Corporation

CUPL™ is a trademark of Personal CAD Systems, Inc.

## LOGIC REFRESHER COURSE

Minimization of EPLD logic equations is normally performed by sophisticated algorithms that eliminate the need for tedious manual reductions. The sections provided here contain logic reference tables for cases where manual reduction techniques may be desirable.



## Boolean Algebra

The Sum-of-Product architecture used in EPLDs makes Boolean algebra ideal for design analysis. The following tables summarize standard Boolean functions.

### Properties

$A * B$	$= B * A$	Commutative Property
$A + B$	$= B + A$	
$A * (B * C)$	$= (A * B) * C$	Associative Property
$A + (B + C)$	$= (A + B) + C$	
$A * (B + C)$	$= A * B + A * C$	Distributive Property
$A + B * C$	$= (A + B) * (A + C)$	

### Postulates

$0 * 0 = 0$	$0 + 0 = 0$	$\bar{0} = 1$
$0 * 1 = 0$	$0 + 1 = 1$	$\bar{1} = 0$
$1 * 1 = 1$	$1 + 1 = 1$	

### Theorems

$A * 0 = 0$	$A + 0 = A$	$\bar{\bar{A}} = A$
$A * 1 = A$	$A + 1 = 1$	
$A * A = A$	$A + A = A$	
$A * \bar{A} = 0$	$A + \bar{A} = 1$	

### DeMorgan's Theorems

$\overline{(A + B + C + D)}$	$=$	$\bar{A} * \bar{B} * \bar{C} * \bar{D}$
$\overline{(A * B * C * D)}$	$=$	$\bar{A} + \bar{B} + \bar{C} + \bar{D}$

### Logic Functions

$A * A$	$=$	<b>A AND A</b>
$A + A$	$=$	<b>A OR A</b>
$\bar{A}$	$=$	<b>A NOT</b>
$A \oplus B = A \text{ EXCLUSIVE OR } B$	$=$	$A\bar{B} + \bar{A}B$

## Karnaugh Maps

Graphical representation of data is usually easier to analyze than strings of ones and zeros. The Karnaugh Map techniques take advantage of this capability and provide an important tool to the logic designer.

### Two Variables

		A	
	B	0	1
0		0	2
1		1	3

296032-7

### Three Variables

		AB			
	C	00	01	11	10
0		0	2	6	4
1		1	3	7	5

296032-8

### Four Variables

		AB			
		CD			
		00	01	11	10
00		0	4	12	8
01		1	5	13	9
11		3	7	15	11
10		2	6	14	10

296032-9

### Five Variables

		BC				BC			
		A=0				A=1			
DE		00	01	11	10	00	01	11	10
		00	01	11	10	00	01	11	10
00	00	0	4	12	8	16	20	28	24
01	01	1	5	13	9	17	21	29	25
11	11	3	7	15	11	19	23	31	27
10	10	2	6	14	10	18	22	30	26

296032-10

### Six Variables

		CD				CD			
		B=0				B=1			
EF		00	01	11	10	00	01	11	10
		00	01	11	10	00	01	11	10
A=0	00	0	4	12	8	16	20	28	24
	01	1	5	13	9	17	21	29	25
	11	3	7	15	11	19	23	31	27
	10	2	6	14	10	18	22	30	26
A=1	00	32	36	44	40	48	52	60	56
	01	33	37	45	41	49	53	61	57
	11	35	39	47	43	51	55	63	59
	10	34	38	46	42	50	54	62	58

296032-11

### Flip-Flop Tables

This subsection includes truth tables and excitation tables for the flip-flops supported by EPLDs.

D Truth Table

D	$Q_N$	$Q_{N+1}$
0	0	0
0	1	0
1	0	1
1	1	1

D Excitation Table

$Q_N$	$Q_{N+1}$	D
0	0	0
0	1	1
1	0	0
1	1	1

T Truth Table

T	$Q_N$	$Q_{N+1}$
0	0	0
0	1	1
1	0	1
1	1	0

T Excitation Table

$Q_N$	$Q_{N+1}$	T
0	0	0
0	1	1
1	0	1
1	1	0

JK Truth Table

J	K	$Q_N$	$Q_{N+1}$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

JK Excitation Table

$Q_N$	$Q_{N+1}$	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

SR Truth Table

S	R	$Q_N$	$Q_{N+1}$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	Illegal	

JK Excitation Table

$Q_N$	$Q_{N+1}$	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

**NOTES:**

$Q_N$  = Present State  
 $Q_{N+1}$  = Next State  
 X = Don't Care

## AUTOMATIC STANDBY MODE (TURBO BIT)

Most Intel EPLDs contain a programmable bit, the Turbo Bit, that optimizes devices for speed or power savings. When TURBO = ON, EPLDs are optimized for speed. When TURBO = OFF, they are optimized for power savings by automatically entering standby

mode when input or I/O transitions are not detected over a short period of time. The following paragraphs describe how the Turbo Bit affects power and speed in EPLDs.

## Turbo Off (Low Power)

Most Intel EPLDs contain circuitry that monitors inputs and feedbacks for transitions. When a transition is detected while the device is in standby mode, the circuit generates an active pulse. The leading edge of this pulse wakes the device up and the device responds according to its programming, changing outputs as necessary. If no new transitions occur during the active pulse, the device enters standby mode again. Outputs are always held valid in standby mode. Input transitions that occur during the active mode interval retrigger the active pulse. The active pulse is different depending on the device (5C060, 5AC312, etc), but is typically several times the propagation delay for a particular device.

In applications with infrequent input transitions, standby mode can result in significant power savings (see the appropriate data sheet for standby power vs. active power). The slight speed loss associated with waking up a device is in the range of 0–30 ns, which is small enough to allow standby mode to be used with most applications (see the appropriate data sheet for effect of Turbo Bit on performance).

## Turbo On (Faster Speed)

In cases where the slight speed loss associated with waking a device from standby mode cannot be traded off to save power, the Turbo bit can be enabled for maximum speed operation. With the Turbo Bit enabled, the device is always in active mode, thus avoiding the wakeup delay. Note that data sheet performance is specified with the Turbo Bit enabled.

The Turbo Bit is enabled/disabled via a TURBO = ON or TURBO = OFF statement in a PLDshell Plus PDS or AIDF option statements. It can also be enabled/disabled by editing the JEDEC file using device programming software. With TURBO = ON the device will be programmed for high speed; with TURBO = OFF the device will be programmed for automatic standby (power savings). The default (erased) state is OFF.



## PAL\*/GAL\* TO PLD REPLACEMENT

Already in wide use throughout the electronics industry are numerous different Programmable Logic Devices. Most common PALs and GALs can be replaced or upgraded with the following Intel PLDs:

### 85C220

The 85C220 is a direct, drop-in replacement for most 20-pin PALs/GALs, although some PALs have an incompatible architecture. The 85C220 runs at up to 100 MHz with external feedback.

### 85C224

The 85C224 is a direct, drop-in replacement for most 24-pin PALs/GALs, although some PALs have an incompatible architecture. The 85C224 runs at up to 100 MHz with external feedback.

### 85C060/PLD610

The 85C060/PLD610 is NOT a drop-in replacement for any 24-pin bipolar PAL, though it can functionally replace many higher-density devices. Some modification of CLK and OE signals may be required.

### 85C22V10/PLD22V10

The PLD22V10 is a direct, drop-in replacement for all 24-pin PALs/GALs. The 85C22V10 includes superset features such as Invertible CLK and expanded feedback options.

\*PAL is a registered trademark of Advanced Micro Devices.  
\*GAL is a registered trademark of Lattice Semiconductor, Incorporated.

### 85C220 As a 20-Pin PAL Replacement

100% Compatible	
10H8, -2	16R6A
12H6, -2	16R4A
14H4, -2	16L8A
16H2, -2	16RP6A
10L8, -2	16RP4A
12L6, -2	16P8A
16L8, A-2, A-4	16R8A
16R4, A-2, A-4	16RP8A
14L4, -2	16V8A
16L2, -2	18P8
16R8, A-2, A-4	18V8
16R6, A-2, A-4	
16P8, -2	
16RP8, -2	
16RP6, -2	
16RP4, -2	
16V8	

### 85C224 As a 24-Pin PAL Replacement

100% Compatible	
14L8	20L8A
16L6	20R8A
18L4	20R6A
20L2	20R4A
20L8	20V8
20R8	
20R6	
20R4	

### 85C060/iPLD610 As a 24-Pin PAL Replacement

Modified Replacement
20RA10
22V10
32V10
26V10
26V12

### 85C22V10/PLD22V10 As a 24-Pin PAL Replacement

100% Compatible
22V10
22VP10

## QUALITY/RELIABILITY

Intel EPLDs meet the same Quality and Reliability standards as Intel's microprocessors and memories. Reliability is not just tested, but is designed into each component Intel manufactures. This assures you that Intel EPLDs will meet your system's quality/reliability needs through its life.

The methods used to guarantee the quality and reliability of Intel EPLDs parallels the methods used with Intel EPROMs. Intel's *Component Quality/Reliability Handbook*, together with *Reliability Report RR-35, EPROM Reliability Data Summary*, can provide the generic information needed to assess Intel's design and testing procedures for EPLDs. Current quality data for specific EPLDs is published in RR-64, *Intel PLD Quality/Reliability Data Summary*, printed in this handbook.

## PACKAGING

Intel EPLDs are available in several packages to meet the wide requirements of customer applications. Current information on available packages is available from your local Intel field sales engineer. Detailed information on package dimensions, etc. for a particular package is provided in *Packaging Outlines and Dimensions*, Order Number 321369, which covers all Intel packages.

## SOFTWARE AND PROGRAMMING SUPPORT

Intel provides design software and programming equipment to support its PLDs. In addition, Intel PLDs are supported by all major compiler and programmer manufacturers. Refer to the appendix to this handbook for detailed information.

1

## ORDERING INFORMATION

### Hotline

The Intel EPLD Technical Hotline is manned by application personnel every business day. The number for the United States and Canada is 1-800-323-EPLD (1-800-323-3753). Outside of the U.S. and Canada, contact your local Intel Sales Office. The Hotline is provided to assist with technical questions concerning Intel EPLDs. A recorder is connected for receiving messages during off-hours or when all applications personnel are busy handling calls.

### BBS

Intel has a Bulletin Board System for registered PLDshell Plus customers to electronically transfer information. Any registered PLDshell Plus user with a modem can log onto the system. The current number is (916) 985-2308. If your communication software supports file transfers, you can receive utilities, software updates, and the latest information on EPLDs via the Bulletin Board.

Data format for the BBS is as follows:

Start Bits: 1

Stop Bits: 1

Data Bits: 8

Speed: 300 or 1200 BAUD

Transmit/receive protocols supported are:

ASCII

XMODEM

KERMIT

TELINK

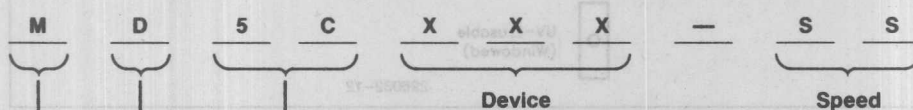
Cyclic Redundancy on XMODEM

### EPLD Design Support

Intel has hardware designers who can help you with your EPLD designs. For more information on design assistance, contact your local Intel field sales office or the EPLD Technical Hotline.

## ORDERING INFORMATION (Continued)

Intel PLDs are identified as follows:

**Family**

- FX — FLEXlogic FPGA (Field Programmable Gate Array)
- PLD — Industry Standard PLD
- 5AC — Advanced Architecture PLD
- 5C — Standard Architecture PLD

**Package Type**

- D — Hermetic, Type D (Cerdip) Dip
- N — Plastic, Leaded Chip Carrier
- P — Plastic Dip and Plastic Flatpack
- X — Unpackaged Device

- A — Indicates automotive operating temperature range ( $-40^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$ )
- J — Indicates a JAN qualified device, but is for internal identification purposes only. All JAN devices must be ordered by M38510 part number. (Example: M38510/42001 BQB), and will be marked in accordance with MIL-M-38510 specifications.
- L — Indicates extended operating temperature range ( $-40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$ ) express product with 160 + 8 hrs. dynamic burn-in.
- \*M — Indicates military operating temperature range ( $-55^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$ )
- Q — Indicates commercial temperature range ( $0^{\circ}\text{C}$  to  $70^{\circ}\text{C}$ ) express product with 160 + 8 hrs. dynamic burn-in.
- T — Indicates extended temperature range ( $-40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$ ) express product without burn-in.
- No letter indicates commercial temperature range ( $0^{\circ}\text{C}$  to  $70^{\circ}\text{C}$ ) without burn-in.

**Examples:**

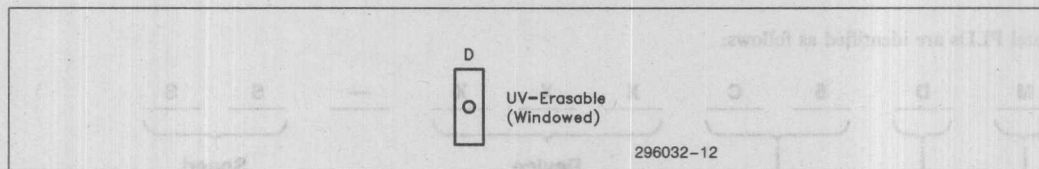
QD5C060-45 Commercial with burn-in, ceramic Dip, 060 (600 gate) device, 45 nanosecond.

\*On military temperature devices, B suffix indicates MIL-STD-883C level B processing.

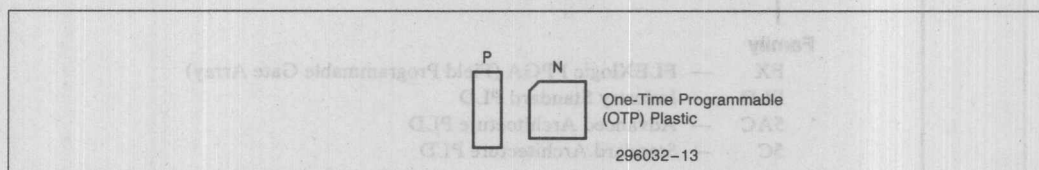
1



## PLD PACKAGE TYPES



D = Windowed Ceramic DIP



P = Plastic DIP  
N = Plastic PLCC

Refer to the Device Summary and each product's data sheet for available packages for a given part.





## EXTENDED FEATURE DEVICES

PLD	Packages	Pins	Mcells (Regs)	Dedic. Inputs	t <sub>PD</sub> (ns)	f <sub>CNT1</sub> (MHz)	t <sub>SU</sub> (ns)	t <sub>CO</sub> (ns)	I <sub>OL</sub> (mA)	I <sub>CC</sub> (mA) @ Freq. (MHz)	Device Type/Application
85C220-80/66	D, P N, M	20 20	8	10	10, 12	80	7	5.5	12	60 @ 80	High-speed, 20-pin PAL/GAL superset (register optimized)
85C220-7/10	N	20	8	10	7.5, 10	74	7	6.5	24	105 @ 74	High-speed, 20-pin PAL/GAL superset (t <sub>PD</sub> optimized)
85C224-80/66	D, P N, M	24, 28	8	14	10, 12	80	7	5.5	12	60 @ 80	High-speed, 24-pin PAL/GAL superset (t <sub>PD</sub> optimized)
85C224-7/10	N	28	8	14	7.5, 10, 15, 25	74	7	6.5	24	105 @ 74	High-speed, 24-pin PAL/GAL superset (register optimized)
85C22V10	D, P N	24 28	10	12	10, 15	71.4	7	7	16	130 @ 15	High-speed, 22V10 superset; Invertible CLK, Expanded Feedback options
85C060	D, P N, M	24 28	16	4	10, 12, 15, 25	74	7	6.5	12	90 @ 1	High-speed, general purpose; 5C060/ EP610/EP630 speed upgrade
85C090	D, P N	40 44	24	12	12, 15, 20, 25	50	11	9	12	150 @ 1	High-speed, general purpose; 5C090/ EP910/EP930 speed upgrade
85C508	D, P N	28 28	8 latches	16	7.5, 10		3 (t <sub>SEU</sub> )	4.5 (t <sub>EO</sub> )	12	25 @ 100	High-speed address decode/latch; active-low outputs
5AC312	D, P N, M	24, 28	12	10	25, 30	33.3	15	15	8	100 @ 33.3	Advanced architecture, general purpose: dual feedback, p-term allocation, 2 Set/Reset OE p-terms, synchronous/ asynchronous clocks
5AC324	D, P N	40, 44	24	12	25, 30	33	12.5	17.8	8	175 @ 33	Advanced architecture, general purpose: dual feedback, p-term allocation, 2 Set/Reset/OE p-terms, synchronous/ asynchronous clocks

**PACKAGES:**

D = Ceramic Dual In-Line Package (Windowed for UV Erase)

N = Plastic J-lead Chip Carrier (One-Time Programmable)

P = Plastic Dual In-Line Package (One-Time Programmable)

M = Military versions available (some speeds/packages); refer to Military Handbook (210461)

\*PAL is a registered trademark of Advanced Micro Devices, Inc.

\*GAL is a registered trademark of Lattice Semiconductor, Corporation.



## HIGH PERFORMANCE INDUSTRY STANDARD DEVICES

PLD	Packages	Pins	Mcells (Regs)	Dedic. Inputs	t <sub>PD</sub> (ns)	f <sub>CNT1</sub> (MHz)	t <sub>SU</sub> (ns)	t <sub>CO</sub> (ns)	I <sub>OL</sub> (mA)	I <sub>CC</sub> (mA) @ Freq. (MHz)	Device Type/Application
PLD16V8XP-7/10	N	20	8	10	7.5, 10	74	7	6.5	24	105 @ 74	High-speed, 20-pin PAL*/GAL* replacement
PLD20V8XP-7/10	N	24, 28	8	14	7.5, 10	74	7	6.5	24	105 @ 74	High-speed, 24-pin PAL/GAL replacement
PLD22V10	P, N	24, 28	10	12	10, 15	71.4	7	7	16	130 @ 15	High-speed, 22V10
PLD610	P, N	24, 28	16	4	10, 15, 25	74	7	6.5	12	105 @ 1	High-speed, general purpose; EP610/EP630 speed upgrade
PLD910	P, N	40, 44	24	12	12, 15, 25	66.7	8	7	12	150 @ 1	High-speed, general purpose; EP910 speed upgrade

## EXTENDED FEATURE DEVICES

PLD	Packages	Pins	Mcells (Regs)	Dedic. Inputs	t <sub>PD</sub> (ns)	f <sub>CNT1</sub> (MHz)	t <sub>SU</sub> (ns)	t <sub>CO</sub> (ns)	I <sub>OL</sub> (mA)	I <sub>CC</sub> (mA) @ Freq. (MHz)	Device Type/Application
5C032	D, P	20	8	10	30, 35, 40	25	23	17	4	30 @ 25	General purpose PLDs; interface logic, state machine sequencers/controllers
5C060	D, P N, M	24 28	16	4	45, 55	16.6	38	22	4	95 @ 96.6	General purpose PLDs; interface logic, state machine sequencers/controllers
5C090	D, P N, M	40 44	24	12	50, 60	16.4	38	23	4	150 @ 16.4	General purpose PLDs; interface logic, state machine sequencers/controllers
5C180	N, M	68	48	16	70, 75, 90	12.1	53	29	4	200 @ 12.2	General purpose PLDs; interface logic, state machine sequencers/controllers

### PACKAGES:

- D = Ceramic Dual In-Line Package (Windowed for UV Erase)
- N = Plastic J-lead Chip Carrier (One-Time Programmable)
- P = Plastic Dual In-Line Package (One-Time Programmable)
- M = Military versions available (some speeds/packages); refer to Military Handbook (210461-009)

\*PAL is a registered trademark of Advanced Micro Devices, Inc.

\*GAL is a registered trademark of Lattice Semiconductor, Corporation.

- High Performance FPGA (Field Programmable Gate Array)
  - Deterministic 10 ns Pin-to-Pin Propagation Delays
  - 80 MHz System Clock Frequency
- 5,000 Equivalent Logic Gates or up to 10,240 Bits of SRAM
- 0.8 $\mu$  CHMOS\* Technology
  - Power Management Options
  - Minimize Active Power Consumption (1.5 mA/MHz)
  - Zero Power Standby
- JTAG 1149.1 Compatible Test Port
  - Supports Boundary Scan and In-circuit Reconfiguration/Programming
- Eight Configurable Function Blocks (CFBs) Linked by a 100% Connectable Matrix
  - Improves Fitting of Complex Designs

- Any CFB can be either 24V10 Logic or SRAM Block
  - Up to 80 Complex Macrocells
  - 128 x 10 SRAM Configuration
  - CFB Selectable 3.3V or 5V Outputs
  - Open-Drain Output Option
- 24V10 Macrocell Features
  - Dual Feedback on All I/O Pins
  - Allocation Supports up to 16 Product Terms Per Macrocell with No Performance Penalty
  - 12 Clocking Options
  - Flexible Preset/Clear Options
  - Selectable D/T/JK/SR Flip-Flops
  - Fast 12-Bit Identity Compare Option
- Supported by Industry Standard Design and Programming Tools

2



290459-1



290459-2

## Package Options

Pins	Package	Macrocells	I/O	Inputs	Clock	JTAG/V <sub>pp</sub>	V <sub>CC</sub>	GND
84	PLCC	80	60	0	2	5	8	9
132	PQFP	80	80	22	2	5	10	13

\*CHMOS is a patented process of Intel Corporation.

## INTRODUCTION

The iFX780 is the first member of the Intel FLEXlogic FPGA (Field Programmable Gate Array) family. The iFX780 consists of eight configurable function blocks (CFBs) linked by a 100% connectable matrix. Each CFB can be defined either as a 24V10 logic block or as a block of 128 x 10 SRAM. This combination will provide approximately 5,000 gates of logic in either PLCC or PQFP packages.

## Flexible Performance

The iFX780 uses Intel's 0.8 $\mu$  CMOS EPROM technology to provide an 80 MHz external clock frequency with predictable 10 ns pin-to-pin delays. This advanced process technology combined with power management options enables very low active and standby power consumption.

## Flexible Features

The unique combination of features available in the iFX780 make it ideal for a wide variety of applications. For example, the high performance and flexible clock options provided are designed to support functions such as bus control, custom cache control, and DRAM control for the current and next generation of Intel microprocessors. The very low power consumption and user selectable 5V/3.3V outputs allow the iFX780 to be used in mixed voltage applications such as portable or embedded systems where CPUs operating at 3.3V still need to communicate to 5V peripherals. The combination of SRAM and logic in a single device becomes a big advantage when designing communication controllers or bus interface controllers where memory is required for buffering data in addition to the logic for the controller design itself.

## Flexible Testing and Programming

The iFX780 also provides dedicated JTAG 1149.1 compatible pins to support boundary scan, in-circuit reconfiguration, and programming modes. In-circuit reconfiguration not only allows the designer ultimate flexibility in prototyping new designs, but also supports applications where the final configuration is not fixed. New configurations may be downloaded to the iFX780 upon power-up to reflect changes in system organization or design requirements that cannot be determined at production time.

## Flexible Tools Support

The FLEXLogic FPGA family is supported by industry standard design entry/programming environ-

ments including Intel's PLDshell Plus™ software. This software runs on i386™ or higher PC-compatible platforms.

## INTERCONNECT

The Global Interconnect Matrix that connects each of the CFB blocks is 100% connectable. Any combination of signals in the matrix can be routed into any CFB block, up to the maximum fan-in of the block (24).

This high degree of connectivity between CFB blocks eliminates routing problems during rework of a complex design.

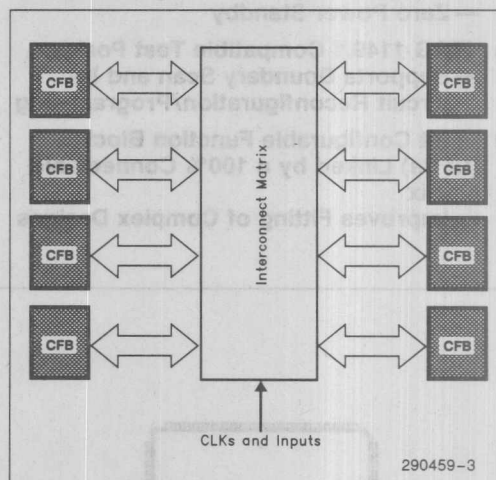


Figure 1. Interconnect Matrix

## CONFIGURABLE FUNCTION BLOCKS

### 24V10 Mode

Each 24V10 block contains a product term array, a P-Term Allocation circuit, 10 macrocells, clocks and I/O logic in the familiar architecture of a simple PLD.

The 24V10 CFB blocks have a superior fan-in to macrocell ratio (2.4:1). This improves the fitting capacity of the iFX780 architecture by providing more available interconnect lines from the global interconnect matrix for each macrocell.

The 24V10 blocks also provide two asynchronous Clear/Preset control terms and two Output Enable control terms (with an inversion option for each). Within each 24V10 block an identity compare circuit is available that can perform a compare of up to 12 bits within the  $t_{PD}$  of the device.

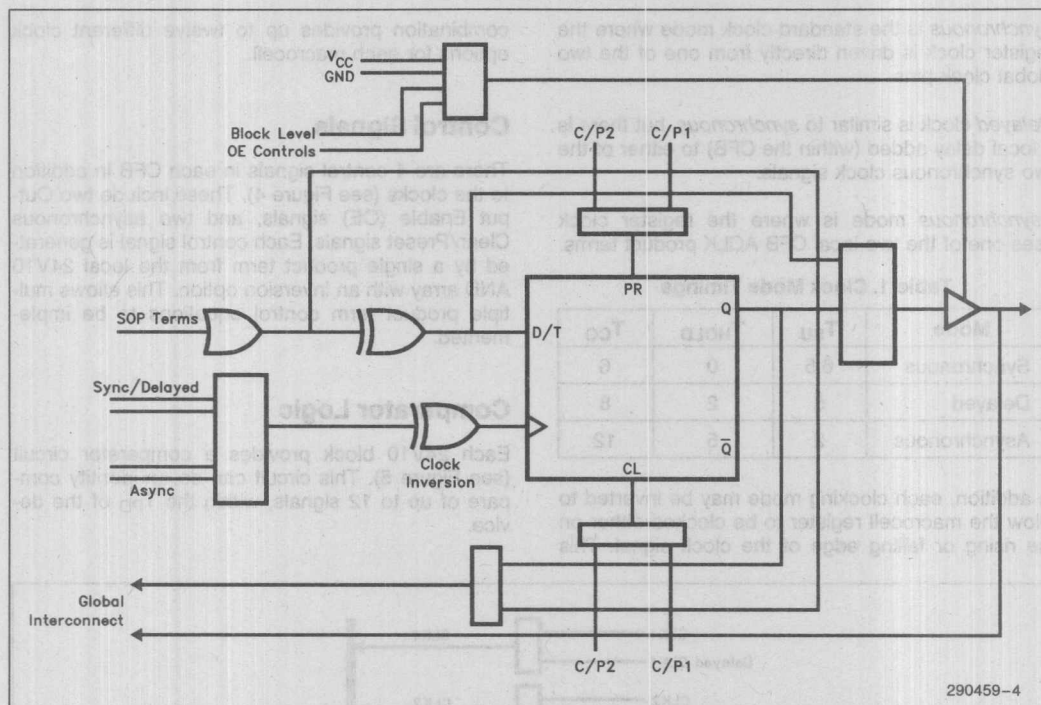


Figure 2. CFB as 24V10 Block

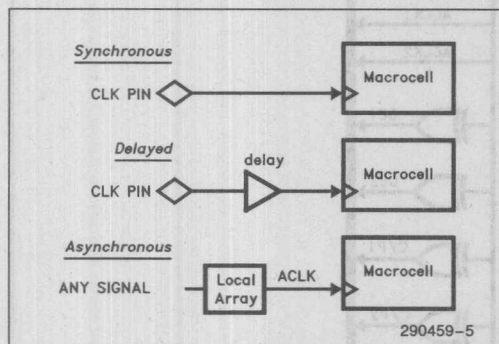


Figure 3. Clocking Modes

## Macrocell Configurations

Each I/O of the device has dual (internal and pin) feedback paths shown in Figure 2. This allows macrocells to be used for buried logic while the I/O pins are used as inputs. Depending on the package used, some macrocell outputs may not be brought outside the package. These I/Os may still be used to provide buried logic since internal feedback is available. The macrocells can be configured either as a fast combinatorial block, a D-register, or a T-register. J/K and S/R registers are available as software emulations.

## Clocking Modes

There are three clocking modes available for every macrocell (see Figure 3): *synchronous*, *delayed*, and *asynchronous*. Table 1 shows the different timing options each clock mode offers.



*Synchronous* is the standard clock mode where the register clock is driven directly from one of the two global clock pins.

*Delayed* clock is similar to *synchronous*, but there is a local delay added (within the CFB) to either of the two synchronous clock signals.

*Asynchronous* mode is where the register clock uses one of the two local CFB ACLK product terms.

**Table 1. Clock Mode Timings**

Mode	T <sub>SU</sub>	T <sub>HOLD</sub>	T <sub>CO</sub>
Synchronous	6.5	0	6
Delayed	5	2	8
Asynchronous	2	5	12

In addition, each clocking mode may be inverted to allow the macrocell register to be clocked either on the rising or falling edge of the clock signal. This

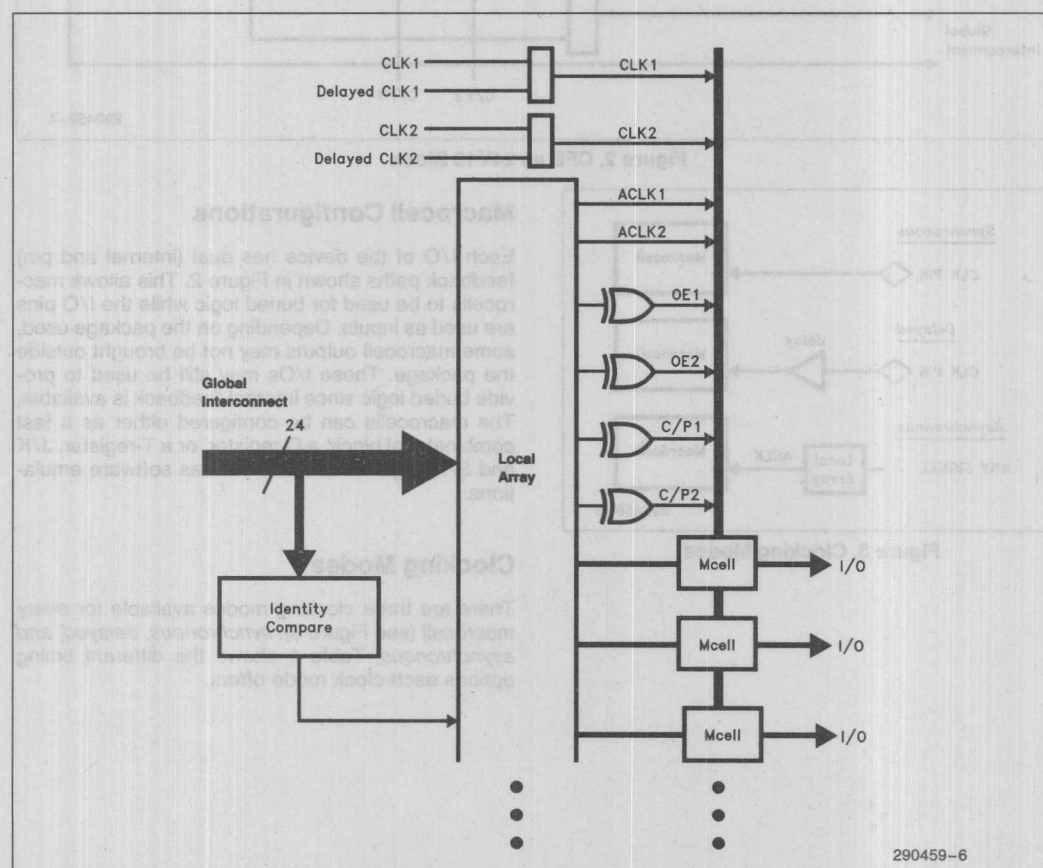
combination provides up to twelve different clock options for each macrocell.

## Control Signals

There are 4 control signals in each CFB in addition to the clocks (see Figure 4). These include two Output Enable (OE) signals, and two asynchronous Clear/Preset signals. Each control signal is generated by a single product term from the local 24V10 AND array with an inversion option. This allows multiple product term control equations to be implemented.

## Comparator Logic

Each 24V10 block provides a comparator circuit (see Figure 5). This circuit can do an identity compare of up to 12 signals, within the T<sub>PD</sub> of the device.



**Figure 4. Control Signals**

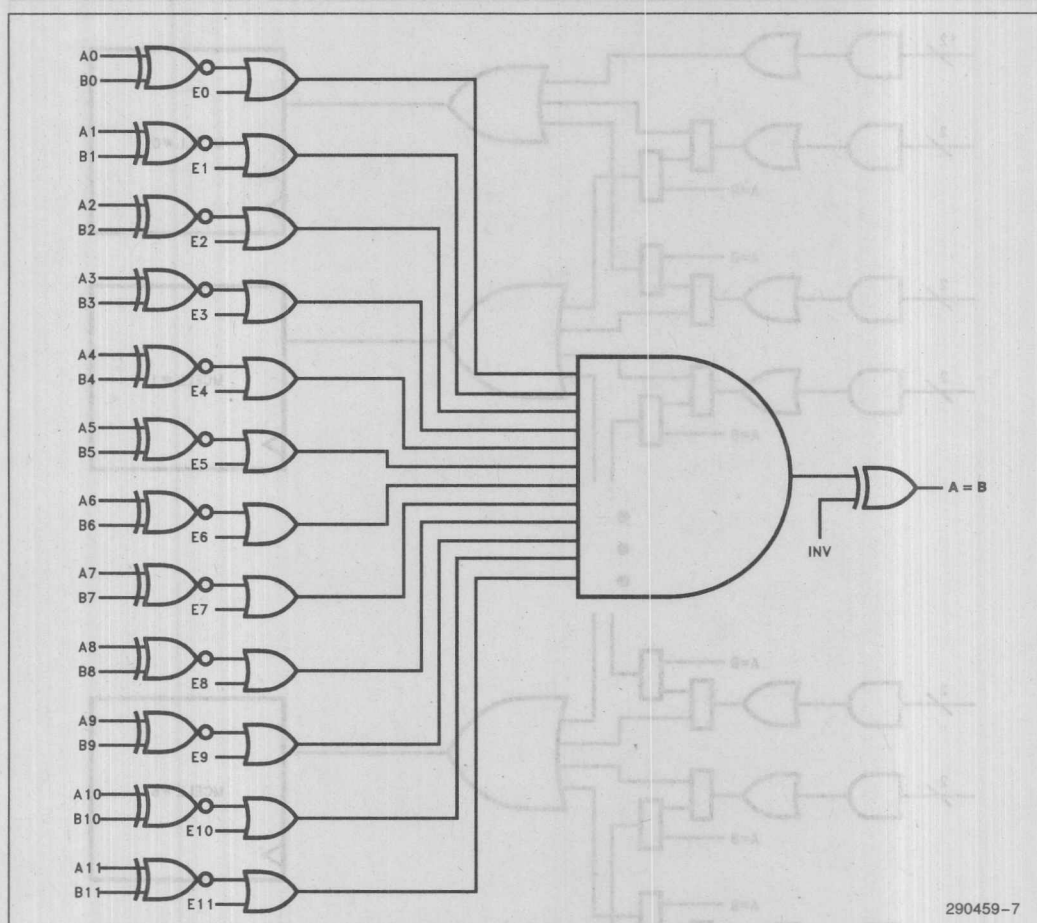


Figure 5. 12-Bit Identity Compare Logic

The number of bits that can be compared is only limited by the fan-in of the CFB. Since the fan-in is 24 signals, a 12-bit comparator is the maximum implementation possible.

When less than 12 bits are being compared, the other signals available from the Interconnect Matrix can be used in equations. For instance, doing an 8-bit compare still leaves 8 other signals on the block fan-in ( $24 - 16 = 8$ ). The bits being compared may also be used to implement SOP logic in parallel with the compare function.

The output of the comparator circuit may be used in place of any of the product term pairs in the product term allocation logic allowing the compare result to be used in any macrocell. However, only one of the ten macrocells in the CFB can use the comparator output.

## Product Term Allocation

The iFX780 uses the patented Intel product term allocation scheme, which gives better utilization of the P-term resources without the performance penalty of other approaches. The P-terms are typically grouped into sets of two product terms each, and there are two sets per macrocell.

Each macrocell may borrow from adjacent macrocells in order to increase the total number of P-terms to a maximum of 8. In addition, the macrocells located at the "ends" of each CFB have access to additional product terms and can support up to 16 P-term equations (see Figure 6). The performance of any macrocell is the same whether 2 or 16 P-terms are being used.

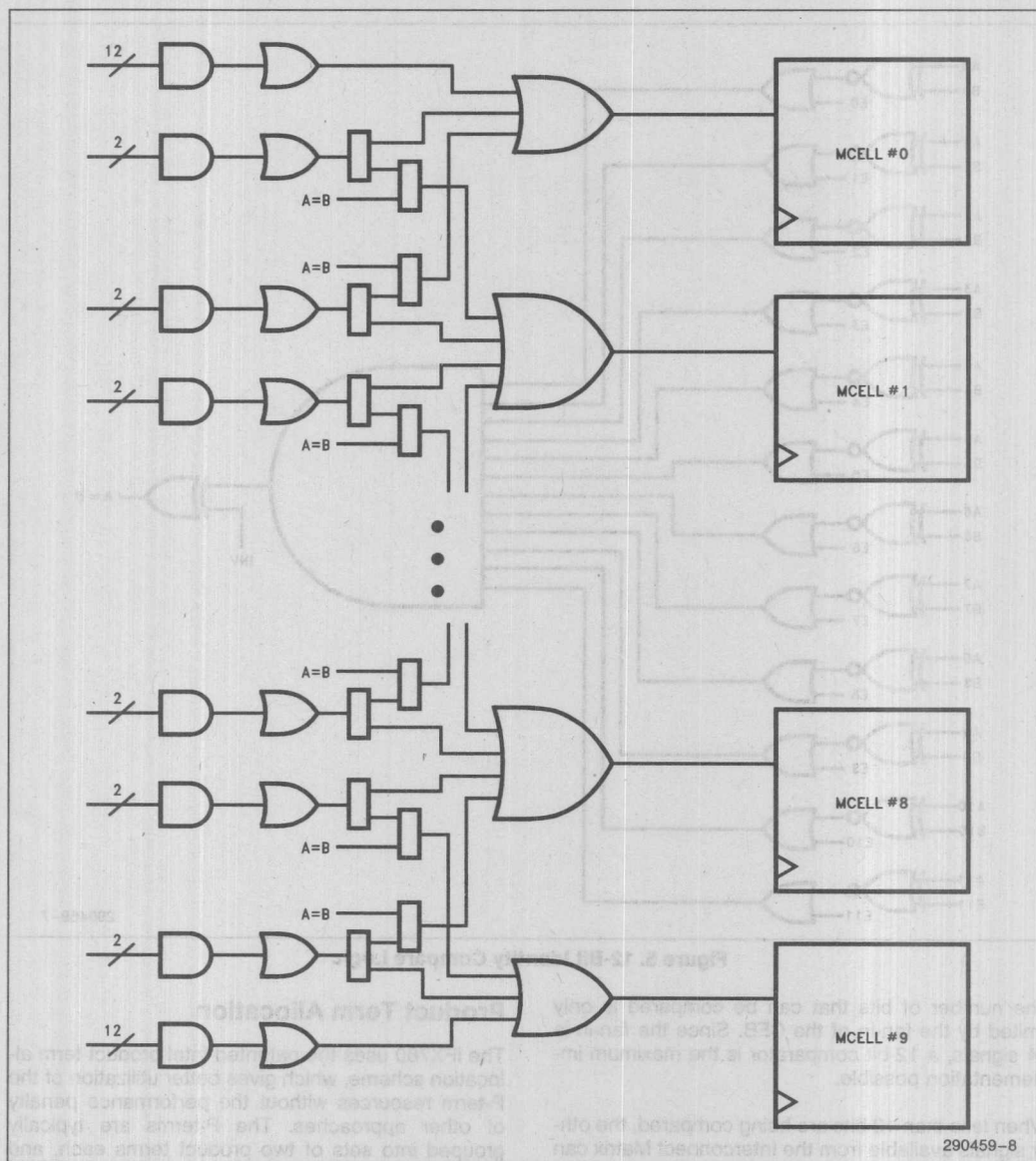


Figure 6. CFB Product Terms

Each macrocell may borrow from adjacent macrocells in order to increase the total number of product terms to a maximum of 8. In addition, the macrocell local-18 P-term equations (see Figure 8). The bottom-18 P-term equations are the same whether 1 or 18 P-terms are being used.

The number of bits that can be combined in the CFB is limited by the CFB. Since the CFB is the maximum number of bits that can be combined in the CFB, the CFB is limited to 18 bits. The CFB is limited to 18 bits.

The output of the comparator circuit may be used in place of any of the product term pairs in the product term allocation logic allowing the compare result to be used in any macrocell. However, only one of the macrocells in the CFB can use the comparator output.

## SRAM Configuration

Each iFX780 CFB block can be configured as a 128x10 (128 words by 10 bits) SRAM block (see Figures 7 and 8). The SRAM is accessed in a conventional manner by using 7 bits of the 24 signal fan-in as address information and 10 bits as data-in. Three bits are used for BE, WE, and OE controls (see Table 2).

Table 2. SRAM Function Table

Inputs			Cycle	I/O Pins
BE	WE	OE		
1	X	X	None	Disabled
0	1	1	Read	Disabled
0	1	0	Read	Enabled
0	0	1	Write	Disabled
0	0	0	Write	Enabled

It is possible to define the SRAM memory either with a bidirectional I/O data bus or with a separate input data bus and output data bus.

The SRAM memory bits are initialized by the on-chip non-volatile configuration cells during power-up. Therefore, the data in the SRAM can be pre-configured at programming time. As long as no memory writes to this block are executed, the SRAM will contain a copy of the nonvolatile cells. In this way, the SRAM block can be used as read only memory (ROM).

When a CFB is configured as a SRAM, regular Sum-of-Product logic is unavailable in that block. All of the macrocells and P-terms have been converted to SRAM use.

Different sized SRAM organizations are possible by cascading multiple CFBs to increase the width or depth of the memory.

## Input Configuration

The iFX780 can be configured to enable a weak feedback pullup option on any CFB input. This option can be used to reduce power consumption for 5V inputs but may increase leakage currents during input transitions.

## Output Configuration

### 3.3V SELECTION

The pins in an I/O block can operate at 3.3V by tying the appropriate  $V_{CC0}$  pins to a 3.3V power supply. While the iFX780 still requires 5V  $V_{CC}$  for normal operation, the  $V_{CC0}$  pin associated with each CFB

block may be connected to either 5V or 3.3V to control the output voltages of the I/O pins in that block. This allows the iFX780 to be used in mixed voltage systems. For example, the iFX780 device may be used as an interface to bridge between a 3.3V CPU and 5V peripheral logic. In addition, all input pins are 5V safe so mixing 3.3V outputs and 5V inputs is supported.

Power sequencing *is* required when any or all CFBs operate at 3.3V levels. In this case, the 5V source must be equal to or greater than the 3.3V source during power-up. During power-down, the 3.3V source must be less than or equal to the 5V source.

## Open Drain Output Option

The device can also be configured to enable an open drain output option for each I/O pin. If desired, more complex equations can be implemented by using multiple open drain outputs with an externally supplied pull-up resistor to emulate an additional OR plane.

## TTL versus CMOS Outputs

There is a weak pullup provided for CMOS compatible outputs. This pullup is always active in both 3.3V and 5V modes.

## JTAG/IEEE 1149.1 TESTABILITY

The JTAG/IEEE 1149.1 Standard Boundary Scan architecture is implemented in the iFX780. This feature supports fault isolation testing of board designs at the component level and enhances production testing, field repair, and is ideal for Fault Tolerant applications.

The iFX780 boundary scan support consists of an Instruction Register, a Data Register, scan cells, and associated logic which are accessed through the Test Access Port (TAP). The TAP interface consists of three inputs: Test Mode Select (TMS), Test Data In (TDI) and Test Clock (TCK), and one output: Test Data Out (TDO).

The boundary scan cells of the iFX780 external signals are linked to form a shift register chain for all active pins. This chain provides a path which can be used to shift in test stimulus as well as shift out test response data for inspection.

For example, a continuity test may be performed between two JTAG devices on a circuit board by placing a known value on the output buffers of one device while observing the input buffers of the other device. This same technique may be used to perform simple in-circuit functional testing of the iFX780 for prototyping new system designs.



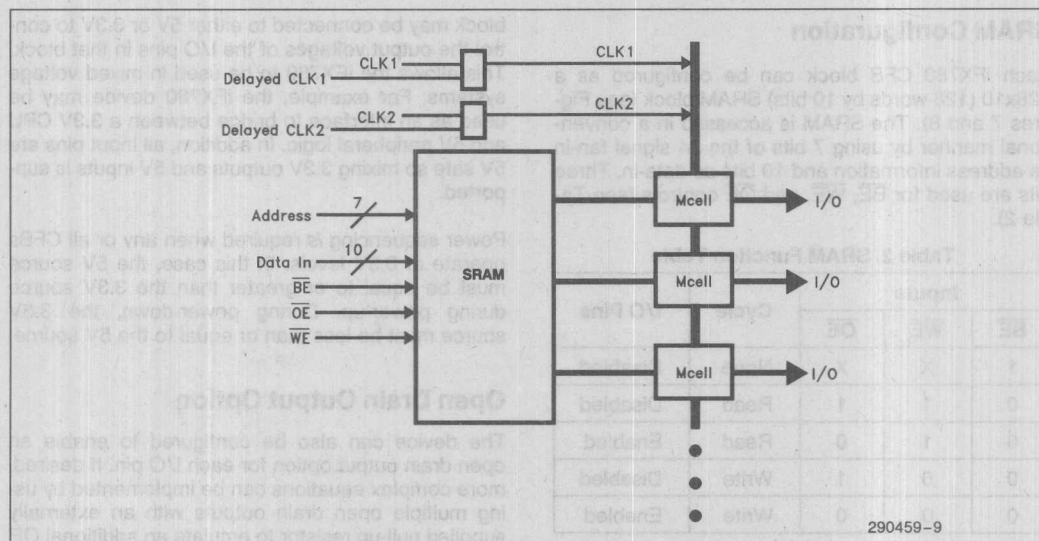


Figure 7. SRAM Overall Block Diagram

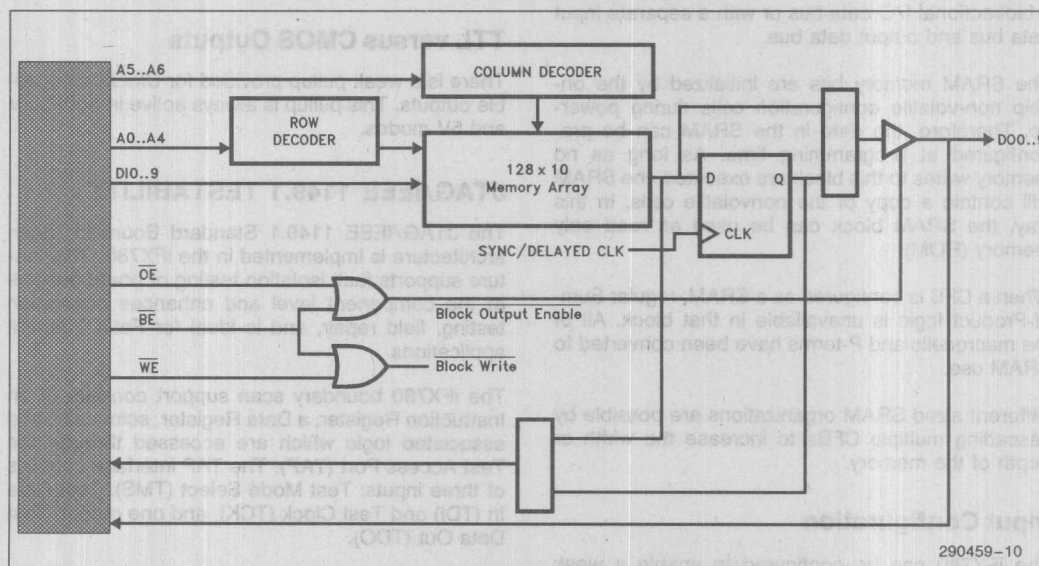


Figure 8. SRAM Functional Block Diagram

The 4-pin JTAG test interface is also used for standard programming, in-circuit reconfiguration, and in-circuit programming.

## Boundary Scan Instructions

The iFX780 boundary scan Instruction Register (IR) supports public instruction opcodes, "semi-public" instruction opcodes used for the Program/Verify modes, and additional Intel private instructions.

## Public Instructions

### EXTEST (IR Opcode 00000 Binary)

The EXTEST instruction drives the output pins to the values contained in the boundary scan cells which allows testing of circuitry external to the iFX780 package, typically for printed circuit board interconnects.

### BYPASS (IR Opcode 11111 Binary)

The BYPASS instruction selects the one bit ByPass Register, (BPR), to be connected to TDI and TDO.

### SAMPLE/PRELOAD (IR Opcode 00001 Binary)

The SAMPLE/PRELOAD instruction is used for two functions. The SAMPLE/PRELOAD instruction 1) allows a "snap-shot" of the values of the pins of the iFX780 in an unobtrusive manner and 2) preloads data to the iFX780 pins to be driven to the system circuit board when executing the EXTEST instruction.

### IDCODE (IR Opcode 00010 Binary)

The IDCODE instruction selects the ID code register to be connected to TDI and TDO allowing the IDcode to be serially shifted out of TDO.

### UESCODE (IR Opcode 10110 Binary)

The UESCODE instruction selects the User Electronic Signature (UES) register to be connected to TDI and TDO allowing the UES code to be serially shifted out of TDO.

### HIZ (IR Opcode 01000 Binary)

The HIZ instruction sets all I/Os to a high impedance state.

## IN-CIRCUIT RECONFIGURATION

The iFX780 supports in-circuit reconfiguration and in-circuit programming through the use of the 4-pin JTAG test port. Downloading a new configuration can be accomplished by simply shifting the new data into the device.

This may be done as many times as desired in a prototyping scenario. Once the final version of the design is confirmed it may be programmed into the non-volatile cells so that the configuration will not be lost even when the power is turned off. This is also done through the use of the JTAG test port plus the programming voltage pin ( $V_{pp}$ ).

For more details on in-circuit reconfiguration and programming please refer to the iFX780 Device Programming and In-Circuit Reconfiguration Specification and supporting application notes.

## SECURITY

A programmable security bit controls access to the data programmed into the device. Once this security bit is set, the design cannot be read out of the non-volatile cells or the SRAM. The state of the nonvolatile security bit at power-up determines access and cannot be changed by in-circuit reconfiguration.

## SOFTWARE SUPPORT

### PLDshell Plus

PLDshell Plus is a sophisticated development tool for Intel programmable logic and is all you need to begin designing with Intel FPGAs. With PLDshell Plus, you can develop, compile, and simulate efficient designs for Intel FPGAs and PLDs.

PLDshell Plus includes several enhancements over earlier versions:

- Design Merge
- SRAM Configuration Support
- Compare Operation Support
- Simulation Support for Intel FPGA
- Vector Notation
- Full Mouse Support
- Device Selector

### Design Merge

PLDshell Plus can merge multiple PDS design files into any Intel PLD, including the Intel iFX780. The Merge function makes it easy for designers to consolidate multiple PLDs into a single, high-performance FPGA or PLD.

## FPGA Architectural Feature Support

PLDshell Plus supports all of the innovative architectural features of the iFX780 through the implementation of new language syntax such as:

- SRAM configuration
- Compare operation
- Buried macrocells
- Clocking options
- 3.3V and 5V options

## Functional Simulation

PLDshell Plus allows the designer to simulate the internal function of any Intel FPGA or PLD for rapid design verification.

PLDshell Plus provides the following simulation capabilities:

- Event-driven simulation of combinatorial, registered, and state machine designs
- Ability to set any input, preload any register, and compare any output against an expected value
- Ability to group signals together (form a vector) to simulate a bus
- Generation of test vectors from simulation results for inclusion in the JEDEC file
- Simulation history file with ability to output a subset of signals to a secondary trace file

## Device Selector

The designer can develop the logic design first, and then use the PLDshell Plus device selector to pick a list of appropriate devices. After a design is compiled or estimated through PLDshell Plus a report file is generated. Contained in the report file is a listing of suggested devices appropriate for the target design.

## System Requirements

Listed below are the minimum requirements for a system in order to use PLDshell Plus:

- Intel 386 based PC compatible
- 2MB RAM (minimum)
- VGA monitor/adaptor
- DOS 3.1 (or later)

## THIRD-PARTY SUPPORT DESIGN SOFTWARE

Third party tools support will be provided by the following vendors:

- Acugen
  - ATGENT™ Test Generation: Automatically generates high coverage functional test vectors for programmable logic devices.
- Cadence
  - Composer™: Comprehensive suite of design entry, debug and documentation capabilities.
  - Verilog-XL™ and VHDL-XL™: Digital logic simulators and interactive debug environment.
- Data I/O
  - ABEL™: Design software allowing you to describe and implement logic designs.
  - PLDtest™ Plus: Integrated software package that combines a testability analysis of the device under design or test with fault grading and automatic test vector generation.
- Logical Devices
  - CUPL™: High level, universal design software package.
- Mentor Graphics
  - Design Architect™: Integrated system of schematic, symbol, and text editors for capturing designs.
  - QuickSim™: High performance logic simulator for function and performance verification.
- Minc
  - PLDesigner-XL(R): Powerful design tool that can be used for all types of programmable logic with automatic device selection, automatic partitioning and functional simulation.
- OrCAD
  - PLD Tools & Schematic Design Tool™: Software tool environment including schematic entry, test vector generation and multiple forms of input.
  - Verification/Simulation Tool™: Series of software tools for performing timing-based simulation of designs.

int\_1

ing setup and hold violations in a design.

- Viewlogic
  - ViewPLD & Powerview™: Integrated schematic capture and simulation environment.

## PROGRAMMING SUPPORT

Programming Support will be provided following vendors:

- BP Microsystems
  - PLD 1100
- Data I/O
  - Unisite 2900/3900
- Elan
  - Model 6000
- Logical Devices
  - ALLPRO
- SMS
  - Sprint Plus

## ADVANCE INFORMATION

Simulation models will be provided by the following vendors:

- Logic Modeling Corporation
  - Smart Model: Device model support for behavioral simulation through a variety of simulators.
- Viewlogic

2

## RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Min	Max
V <sub>DD</sub>	Supply Voltage - V <sub>DD</sub>	4.75	5.25
V <sub>CC</sub>	Output Supply Voltage - V <sub>CC</sub>	3.0	3.0
V <sub>IN</sub>	Input Voltage	0	V <sub>CC</sub>
V <sub>O</sub>	Output Voltage	0	V <sub>CC</sub>
T <sub>A</sub>	Operating Temperature	0	+70
t <sub>PL</sub>	Input Rise Time		500
t <sub>PH</sub>	Input Fall Time		500

## D.C. CHARACTERISTICS (T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5.0V ± 5%)

Symbol	Parameter	Min	Typ	Max	Units	Test Conditions
V <sub>OH</sub>	High Level Input Voltage	2.0		V <sub>CC</sub> + 0.3	V	
V <sub>OL</sub>	Low Level Input Voltage	-0.3		0.3	V	
V <sub>OH</sub>	High Level Output	2.4			V	I <sub>O</sub> = -4.0 mA D.C. V <sub>CC</sub> = Min
V <sub>OH</sub>	High Level Output	V <sub>CC</sub> - 0.2			V	I <sub>O</sub> = -20 mA D.C. V <sub>CC</sub> = Min
V <sub>OH</sub>	High Level Output Voltage	V <sub>CC</sub> - 0.2			V	I <sub>O</sub> = -20 mA D.C. V <sub>CC</sub> = Min
V <sub>OL</sub>	Low Level Output Voltage			0.45	V	I <sub>O</sub> = 20 mA D.C. V <sub>CC</sub> = Min
V <sub>OL</sub>	Low Level Output Voltage			0.2	V	I <sub>O</sub> = 20 mA D.C. V <sub>CC</sub> = Min
I <sub>Q</sub>	Input Leakage Current			±10	nA	V <sub>IN</sub> = (GND) or V <sub>CC</sub> V <sub>CC</sub> = Max



## ABSOLUTE MAXIMUM RATINGS\*

Symbol	Parameter	Min	Max	Units
V <sub>CC</sub>	Supply Voltage <sup>(1)</sup>	-2.0	+7.0	V
V <sub>PP</sub>	Programming Supply Voltage <sup>(1)</sup>	-2.0	+13.5	V
V <sub>I</sub>	DC Input Voltage <sup>(1, 2)</sup>	-0.5	V <sub>CC</sub> + 0.5	V
t <sub>STG</sub>	Storage Temperature	-65	+150	°C
t <sub>AMB</sub>	Ambient Temperature <sup>(3)</sup>	-10	+85	°C

NOTICE: This data sheet contains information on products in the sampling and initial production phases of development. The specifications are subject to change without notice. Verify with your local Intel Sales office that you have the latest data sheet before finalizing a design.

\*WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.

## NOTES:

1. Voltages with respect to ground.
2. Minimum DC input is -0.5V. During transitions, the inputs may undershoot to -2.0V or overshoot to +7.0V for periods less than 20 ns under no load conditions.
3. Under bias. Extended temperature versions are also available.

## RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Min	Max	Units
V <sub>CC</sub> /V <sub>CCO</sub>	Supply Voltage - 5V	4.75	5.25	V
V <sub>CCO</sub>	Output Supply Voltage - 3.3V	3.0	3.6	V
V <sub>IN</sub>	Input Voltage	0	V <sub>CC</sub>	V
V <sub>O</sub>	Output Voltage	0	V <sub>CCO</sub>	V
T <sub>A</sub>	Operating Temperature	0	+70	°C
t <sub>r</sub>	Input Rise Time		500	ns
t <sub>f</sub>	Input Fall Time		500	ns

D.C. CHARACTERISTICS (T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5.0V ± 5%)(4)

Symbol	Parameter	Min	Typ	Max	Units	Test Conditions
V <sub>IH</sub> <sup>(5)</sup>	High Level Input Voltage	2.0		V <sub>CC</sub> + 0.3	V	
V <sub>IL</sub> <sup>(5)</sup>	Low Level Input Voltage	-0.3		0.8	V	
V <sub>OH</sub>	5V TTL High Level Output	2.4			V	I/O = -4.0 mA D.C., V <sub>CC</sub> = Min
	5V CMOS High Level Output	V <sub>CCO</sub> - 0.2			V	I/O = -20 µA D.C., V <sub>CC</sub> = Min
	3V High Level Output Voltage	V <sub>CCO</sub> - 0.2			V	I/O = -20 µA D.C., V <sub>CC</sub> = Min
V <sub>OL</sub>	5V Low Level Output Voltage			0.45	V	I/O = 12.0 mA D.C., V <sub>CC</sub> = Min
	3V Low Level Output Voltage			0.2	V	I/O = 20 µA D.C., V <sub>CC</sub> = Min
I <sub>I</sub>	Input Leakage Current			± 10	µA	V <sub>CC</sub> = Max, V <sub>IN</sub> = GND or V <sub>CC</sub>

## NOTES:

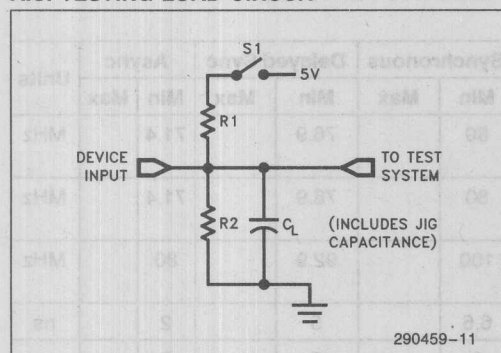
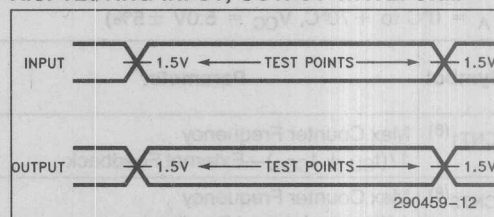
4. Typical values are at T<sub>A</sub> = 25°C, V<sub>CC</sub> = 5V.
5. Absolute values with respect to device GND; all over and undershoots due to system and tester noise are included. Do not attempt to test these values without suitable equipment.

**D.C. CHARACTERISTICS** ( $T_A = 0^\circ\text{C}$  to  $+70^\circ\text{C}$ ,  $V_{CC} = 5.0\text{V} \pm 5\%$ )(4) (Continued)

Symbol	Parameter	Min	Typ	Max	Units	Test Conditions
$I_{OZ}$	Output Leakage Current			$\pm 10$	$\mu\text{A}$	$V_{CC} = \text{Max}$ , $V_{OUT} = \text{GND}$ or $V_{CC}$
$I_{SC}^{(6)}$	Output Short Circuit Current	-30		-120	mA	$V_{CC} = \text{Max}$ , $V_{OUT} = 0.5\text{V}$
$I_{SB}$	Standby Power Supply Current		1		mA	$V_{IN} = V_{CC}$ or GND, Outputs Open
$I_{CC}$ Active	Power Supply Current		1.5		mA per MHz	$V_{IN} = V_{CC}$ or GND, Outputs Open, Device Programmed as Four 20-Bit Counters

**NOTE:**

6. Not more than 1 output should be tested at a time. Duration of that test should not exceed 1 second.

**A.C. TESTING LOAD CIRCUIT****A.C. TESTING INPUT, OUTPUT WAVEFORM**

2

**SWITCHING TEST CIRCUIT**

Specification	S1	$C_L$	Commercial		Measured Output Value
			R1	R2	
$t_{PD}$	Closed	35 pF	330 $\Omega$	200 $\Omega$	1.5V
$t_{PZX}$	Z $\rightarrow$ H: Open Z $\rightarrow$ L: Closed				1.5V
$t_{PXZ}$	H $\rightarrow$ Z: Open L $\rightarrow$ Z: Closed	5 pF			H $\rightarrow$ Z: $V_{OH} - 0.5\text{V}$ L $\rightarrow$ Z: $V_{OL} + 0.5\text{V}$

**PIN CAPACITANCE** ( $T_A = 0^\circ\text{C}$  to  $+70^\circ\text{C}$ ,  $V_{CC} = 5.0\text{V} \pm 5\%$ )(7)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$C_{IN}$	Input Capacitance	$V_{IN} = 2\text{V}$ , $f = 1.0\text{ MHz}$		10	12	pF
$C_{IO}$	I/O Capacitance	$V_{OUT} = 2\text{V}$ , $f = 1.0\text{ MHz}$		12	15	pF
$C_{CLK}$	Clock Pin Capacitance	$V_{OUT} = 2\text{V}$ , $f = 1.0\text{ MHz}$		15	18	pF
$C_{VPP}$	$V_{PP}$ Pin Capacitance	$f = 1.0\text{ MHz}$		12	15	pF

**NOTE:**

7. These values are evaluated at initial characterization and whenever design modifications occur that may affect capacitance.

**COMBINATORIAL MODE A.C. CHARACTERISTICS**(T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5.0V ± 5%)

Symbol	Parameter	iFX780-10			iFX780-15			Units
		Min	Typ	Max	Min	Typ	Max	
t <sub>PD</sub> <sup>(8)</sup>	Input or I/O to Output Valid			10			15	ns
t <sub>PZX</sub> <sup>(9)</sup>	Input or I/O to Output Enable			12			18	ns
t <sub>PXZ</sub> <sup>(9)</sup>	Input or I/O to Output Disable			12			18	ns
t <sub>CLR</sub>	Input or I/O to Asynchronous Clear/Preset			15			20	ns
t <sub>COMP</sub>	Comparator Input or I/O Feedback to Output Valid			10			15	ns

**REGISTER MODE—iFX780-10 CLOCK A.C. CHARACTERISTICS**(T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5.0V ± 5%)

Symbol	Parameter	Synchronous		Delayed Sync		Async		Units
		Min	Max	Min	Max	Min	Max	
f <sub>CNT1</sub> <sup>(8)</sup>	Max Counter Frequency 1/(t <sub>SU</sub> + t <sub>CO1</sub> )—External Feedback	80		76.9		71.4		MHz
f <sub>CNT2</sub> <sup>(8)</sup>	Max Counter Frequency 1/(t <sub>CNT</sub> )—Internal Feedback	80		76.9		71.4		MHz
f <sub>MAX</sub> <sup>(8)</sup>	Max Frequency (Pipelined) 1/(t <sub>CP</sub> )—No Feedback	100		92.9		80		MHz
t <sub>SU</sub>	Input or I/O Setup Time to CLK	6.5		5		2		ns
t <sub>H</sub>	Input or I/O Hold Time from CLK	0		2		5		ns
t <sub>CO1</sub>	CLK to Output Valid		6		8		12	ns
t <sub>CO2</sub>	CLK to Output Valid Fed Through Combinatorial Macrocell		16		18		22	ns
t <sub>CNT</sub>	Register Output Feedback to Register Input— Internal Path		12.5		13		14	ns
t <sub>CL</sub>	CLK Low Time	4		4		5		ns
t <sub>CH</sub>	CLK High Time	4		4		5		ns
t <sub>CP</sub>	CLK Period	10		10.5		12.5		ns

**NOTES:**

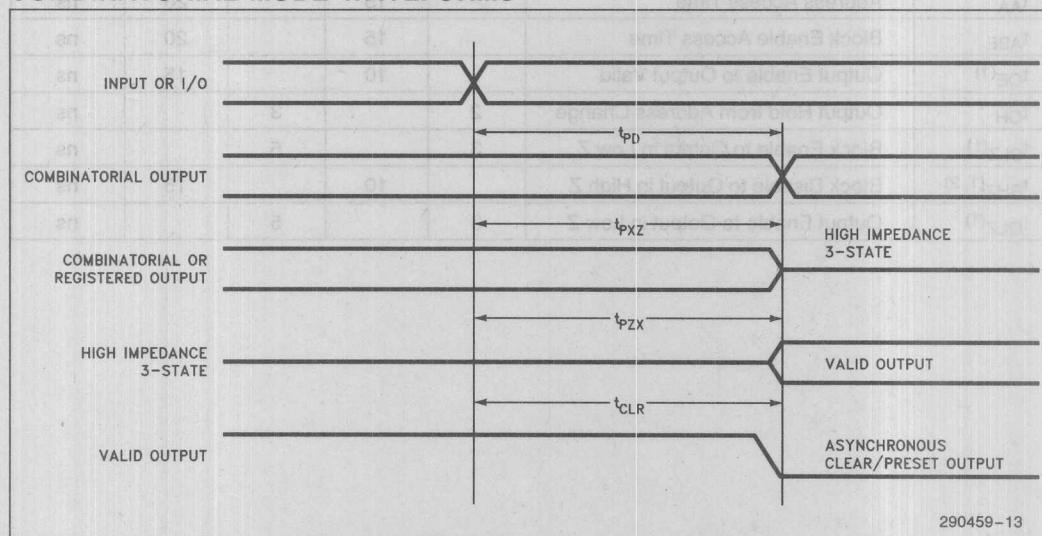
8. Half outputs switching per block.

9. t<sub>PZX</sub> and t<sub>PXZ</sub> are measured at ±0.5V from steady state voltage as driven by specified output load. t<sub>PXZ</sub> is measured with C<sub>L</sub> = 5 pF. Z → H and Z → L are measured at 1.5V on output.

**REGISTER MODE—iFX780-15 CLOCK A.C. CHARACTERISTICS**(T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5.0V ± 5%)

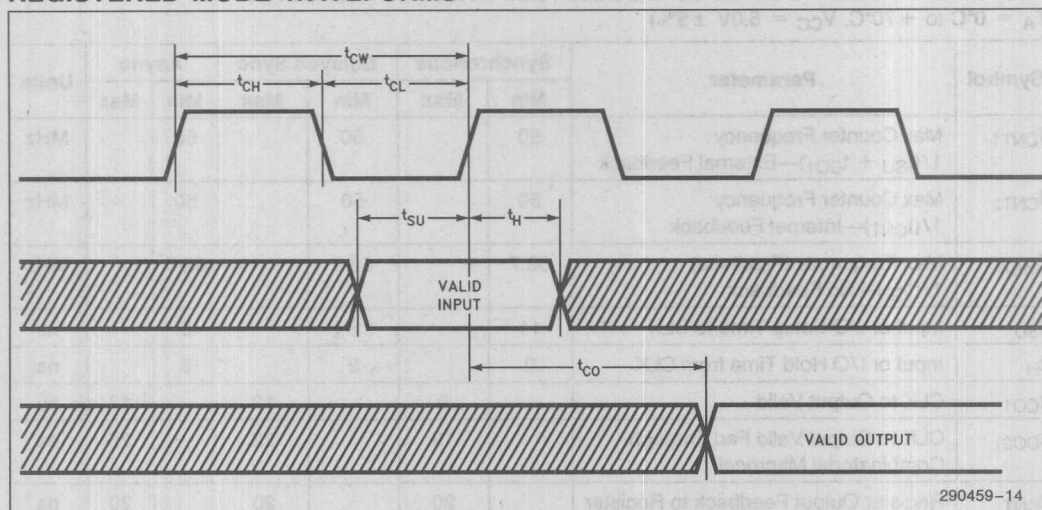
Symbol	Parameter	Synchronous		Delayed Sync		Async		Units
		Min	Max	Min	Max	Min	Max	
f <sub>CNT1</sub>	Max Counter Frequency 1/(t <sub>SU</sub> + t <sub>CO1</sub> )—External Feedback	50		50		50		MHz
f <sub>CNT2</sub>	Max Counter Frequency 1/(t <sub>CNT</sub> )—Internal Feedback	50		50		50		MHz
f <sub>MAX</sub>	Max Frequency (Pipelined) 1/(t <sub>CP</sub> )—No Feedback	66.7		62.5		62.5		MHz
t <sub>SU</sub>	Input or I/O Setup Time to CLK	11		8		3		ns
t <sub>H</sub>	Input or I/O Hold Time from CLK	0		2		6		ns
t <sub>CO1</sub>	CLK to Output Valid		9		12		17	ns
t <sub>CO2</sub>	CLK to Output Valid Fed Through Combinatorial Macrocell		19		22		27	ns
t <sub>CNT</sub>	Register Output Feedback to Register Input—Internal Path		20		20		20	ns
t <sub>CL</sub>	CLK Low Time	7		7		7		ns
t <sub>CH</sub>	CLK High Time	7		7		7		ns
t <sub>CP</sub>	CLK Period	15		15		15		ns

2

**COMBINATORIAL MODE WAVEFORMS**

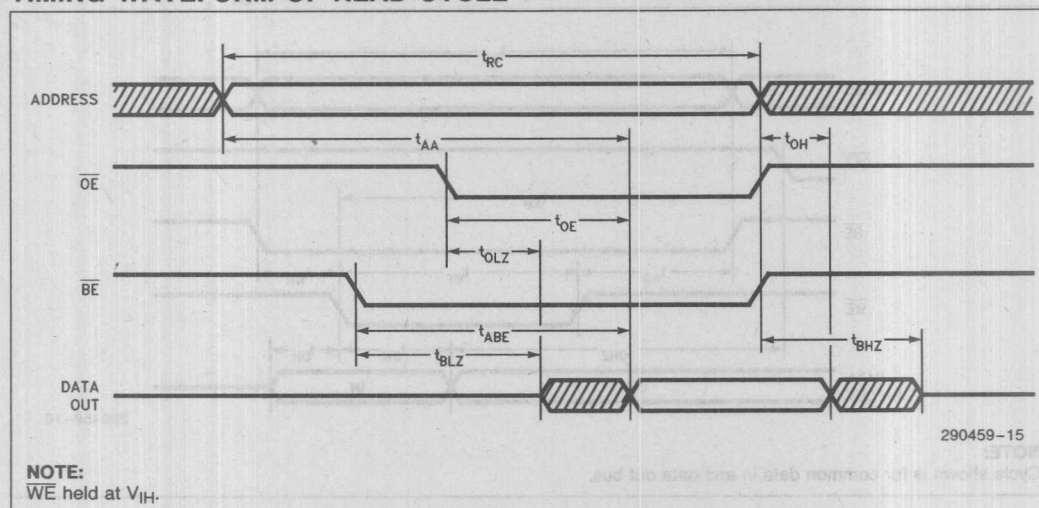


## REGISTERED MODE WAVEFORMS

SRAM READ—A.C. CHARACTERISTICS ( $T_A = 0^\circ\text{C}$  to  $+70^\circ\text{C}$ ,  $V_{CC} = 5.0\text{V} \pm 5\%$ )

Symbol	Parameter	iFX780-10		iFX780-15		Units
		Min	Max	Min	Max	
$t_{RC}$	Read Cycle Time	15		20		ns
$t_{AA}$	Address Access Time		15		20	ns
$t_{ABE}$	Block Enable Access Time		15		20	ns
$t_{OE}^{(1)}$	Output Enable to Output Valid		10		15	ns
$t_{OH}$	Output Hold from Address Change	2		3		ns
$t_{BLZ}^{(1)}$	Block Enable to Output in Low Z	3		5		ns
$t_{BHZ}^{(1,2)}$	Block Disable to Output in High Z		10		15	ns
$t_{OLZ}^{(1)}$	Output Enable to Output in Low Z	3		5		ns

## TIMING WAVEFORM OF READ CYCLE



2

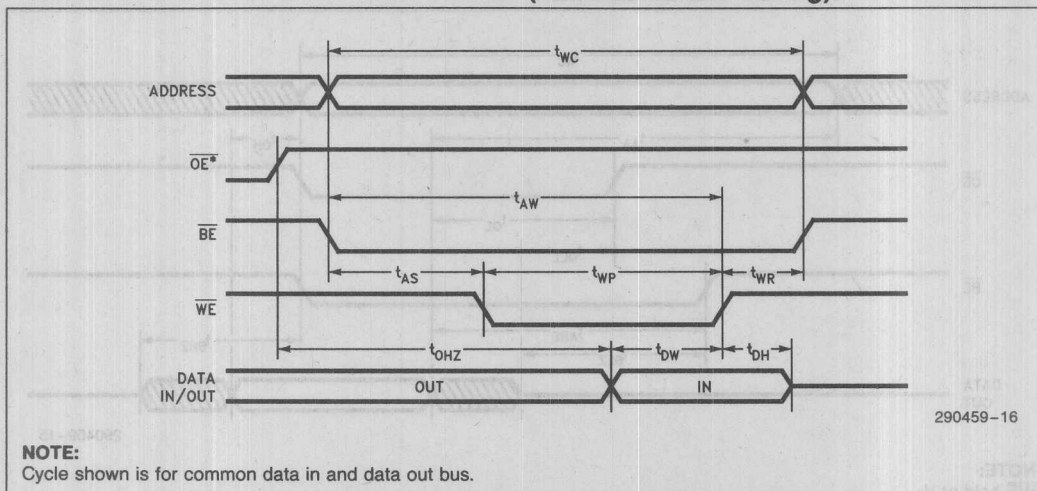
## SRAM WRITE—A.C. CHARACTERISTICS ( $T_A = 0^\circ\text{C}$ to $+70^\circ\text{C}$ , $V_{CC} = 5.0\text{V} \pm 5\%$ )

Symbol	Parameter	iFX780-10		iFX780-15		Units
		Min	Max	Min	Max	
$t_{WC}$	Write Cycle Time	15		20		ns
$t_{BW}$	Block Enable to End of Write	12		16		ns
$t_{AW}$	Address Valid to End of Write	15		20		ns
$t_{AS}$	Address Set-up Time	3		4		ns
$t_{WP}$	Write Pulse Width	12		16		ns
$t_{WR}$	Write Recovery Time	0		0		ns
$t_{DW}$	Data Valid to End of Write	12		16		ns
$t_{DH}$	Data Hold Time	0		0		ns
$t_{OHZ}^{(1, 2, 3)}$	Output Disable to Valid Data In	10		13		ns

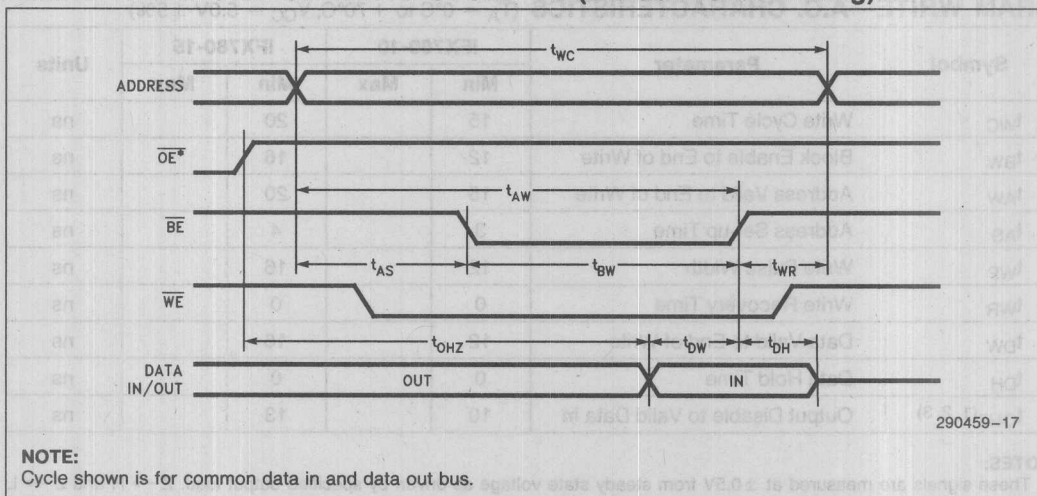
### NOTES:

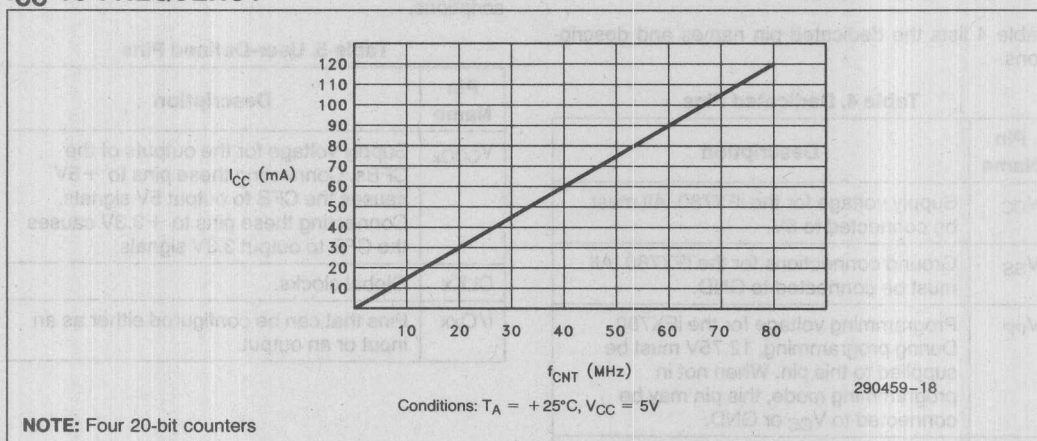
- These signals are measured at  $\pm 0.5\text{V}$  from steady state voltage as driven by specified output load.  $Z \rightarrow H$  and  $Z \rightarrow L$  are measured at  $1.5\text{V}$  on output.
- These signals are measured with  $C_L = 5\text{ pF}$ .
- Does not apply for separate data in and data out buses.

# TIMING WAVEFORM OF WRITE CYCLE ( $\overline{WE}$ Controlled Timing)



# TIMING WAVEFORM OF WRITE CYCLE #2 ( $\overline{BE}$ Controlled Timing)



**I<sub>CC</sub> vs FREQUENCY**

2

**POWER-UP RESET**

Because  $V_{CC}$  rise can vary significantly from one application to another,  $V_{CC}$  rise must be monotonic. The power-up cycle is complete within a delay of  $t_{PR}$  after  $V_{CC}$  reaches the  $V_{ON}$  value.

Internal power-up reset circuits ensure that all flip-flops will be reset to a logic 0 after the device has powered-up. Also, the JTAG TAP controller will be put into the *Test-Logic-Reset* state.

**POWER-UP RESET CHARACTERISTICS**

Symbol	Parameter	Value
$t_{PR}$	Power-Up Reset Time	100 $\mu\text{s}$ Max
$V_{ON}$	Turn-On Voltage	4.75V Min



## PIN DESCRIPTIONS

Table 4 lists the dedicated pin names and descriptions.

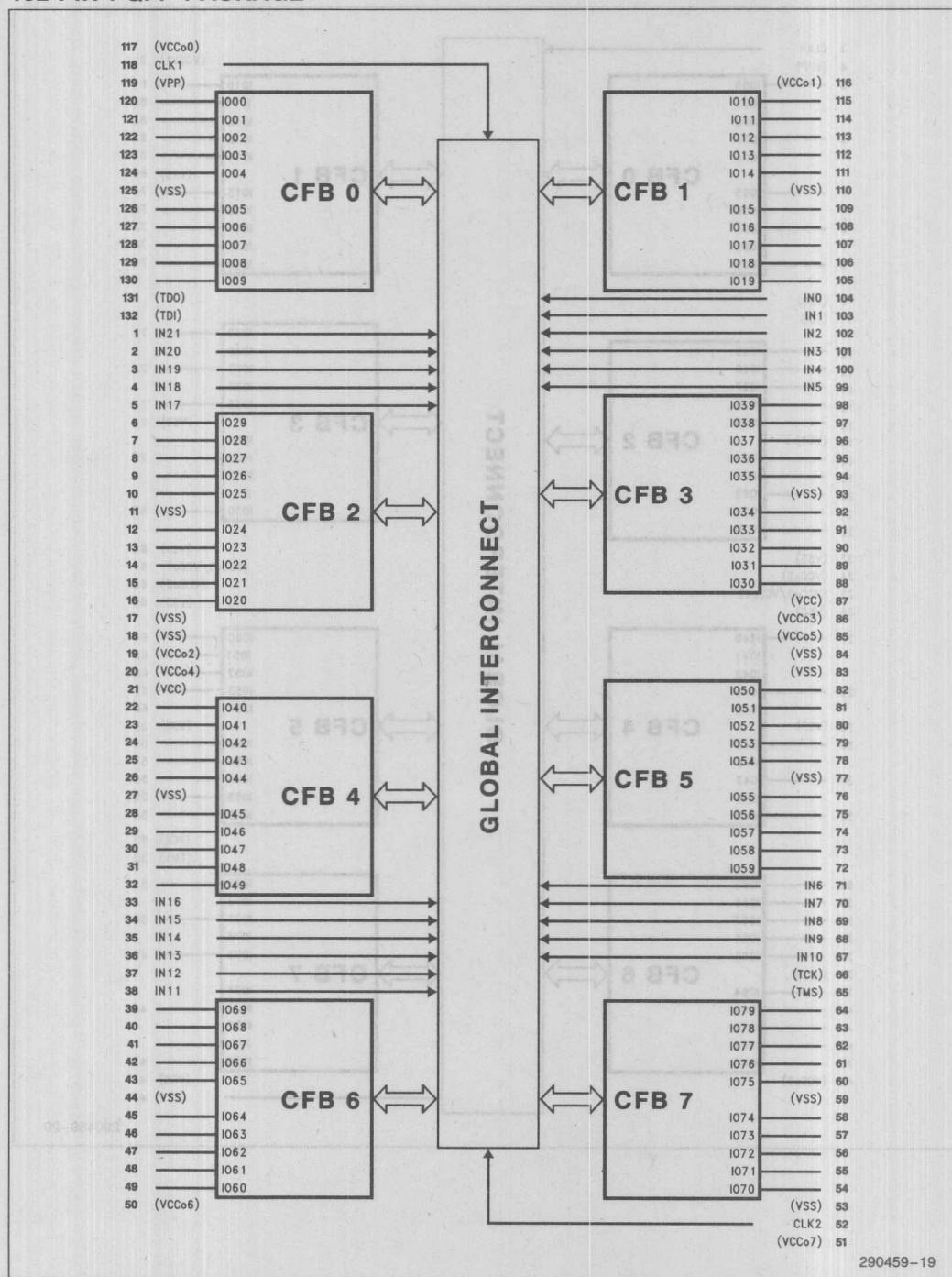
Table 4. Dedicated Pins

Pin Name	Description
V <sub>CC</sub>	Supply voltage for the iFX780. All must be connected to 5V.
V <sub>SS</sub>	Ground connections for the iFX780. All must be connected to GND.
V <sub>PP</sub>	Programming voltage for the iFX780. During programming, 12.75V must be supplied to this pin. When not in programming mode, this pin may be connected to V <sub>CC</sub> or GND.
IN <sub>x</sub>	Input only pins. These pins may not be available on all packages.
TDI	The Testability Data Input is the boundary scan serial data input to the iFX780. JTAG instructions and data are shifted into the iFX780 on the TDI input pin on the rising edge of TCK.
TDO	The Testability Data Output is the boundary scan serial data output from the iFX780. JTAG instructions and data are shifted out of the iFX780 on the TDO output on the falling edge of TCK.
TCK	The Testability Clock input provides the boundary scan clock for the iFX780. TCK is used to clock state information and data into and out of the iFX780 during boundary scan or programming modes. The maximum operating frequency of the boundary scan test clock is 20 MHz.
TMS	The Testability Control input is the boundary scan test mode select for the iFX780.

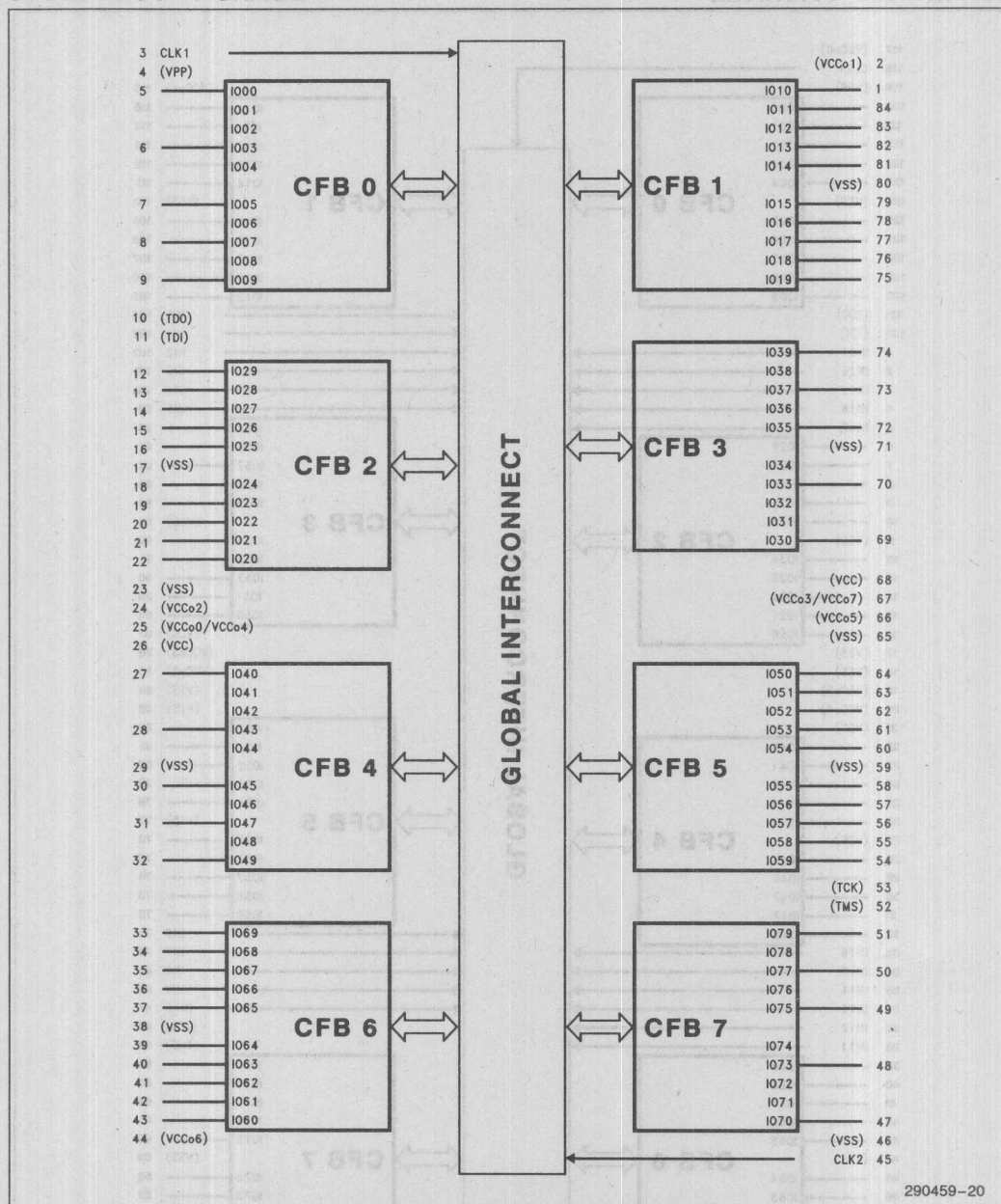
Table 5 lists the user defined pin names and descriptions.

Table 5. User-Defined Pins

Pin Name	Description
V <sub>CCOx</sub>	Supply voltage for the outputs of the CFBs. Connecting these pins to +5V causes the CFB to output 5V signals. Connecting these pins to +3.3V causes the CFB to output 3.3V signals.
CLK <sub>x</sub>	Global clocks.
I/O <sub>xx</sub>	Pins that can be configured either as an input or an output.



# 84-PIN PLCC PACKAGE



290459-20

**iFX740****10 ns FLEXlogic™ FPGA FAMILY WITH SRAM OPTION**

- High Performance FPGA (Field Programmable Gate Array)
  - Deterministic 10 ns Pin-to-Pin Propagation Delays
  - 80 MHz System Clock Frequency
- 2500 Equivalent Logic Gates or up to 5,128 Bits of SRAM
- 0.8 $\mu$  CHMOS\* Technology
  - Power Management Options
  - Minimize Active Power Consumption (0.75 mA/MHz)
  - Zero Power Standby
- Four Configurable Function Blocks (CFBs) Linked by a 100% Connectable Matrix
  - Improves Fitting of Complex Designs
- Any CFB Can Be either 24V10 Logic or SRAM Block
  - Up to 40 Complex Macrocells
  - 128 x 10 SRAM Configuration
  - CFB Selectable 3.3V or 5V Outputs
  - Open-Drain Output Option
- 24V10 Macrocell Features
  - Dual Feedback on All I/O Pins
  - Allocation Supports up to 16 Product Terms per Macrocell with No Performance Penalty
  - 12 Clock Options
  - Flexible Preset/Clear Options
  - Selectable D/T/JK/SR Flip-Flops
  - Fast 12-Bit Identity Compare Option
- Supported by Industry Standard Design and Programming Tools

2

**Package Options**

Pins	Package	Macrocells	I/O	CLK/ Inputs	V <sub>PP</sub> /V <sub>CC</sub> / GND
44	PLCC	40	32	2	10
68	PLCC	40	40	16	12

\*CHMOS is a patented process of Intel Corporation.





## iFX7160

### 10 ns FLEXlogic™ FPGA FAMILY WITH SRAM OPTION

- **High Performance FPGA (Field Programmable Gate Array)**
  - Deterministic 10 ns Pin-to-Pin Propagation Delays
  - 80 MHz System Clock Frequency
- **10,000 Equivalent Logic Gates or up to 20,480 Bits of SRAM**
- **0.8 $\mu$  CHMOS\* Technology**
  - Power Management Options
  - Minimize Active Power Consumption (2.5 mA/MHz)
  - Zero Power Standby
- **JTAG 1149.1 Compatible Test Port**
  - Supports Boundary Scan and In-Circuit Reconfiguration/Programming
- **16 Configurable Function Blocks (CFBs) Linked by a 100% Connectable Matrix**
  - Improves Fitting of Complex Designs
- **Any CFB Can Be Either 24V10 Logic or SRAM Block**
  - Up to 160 Complex Macrocells
  - 128 x 10 SRAM Configuration
  - CFB Selectable 3.3V or 5V Outputs
  - Open-Drain Output Option
- **24V10 Macrocell Features**
  - Dual Feedback on All I/O Pins
  - Allocation Supports up to 16 Product Terms per Macrocell with No Performance Penalty
  - 12 Clock Options
  - Flexible Preset/Clear Options
  - Selectable D/T/JK/SR Flip-Flops
  - Fast 12-Bit Identity Compare Option
- **Supported by Industry Standard Design and Programming Tools**

#### Package Options

Pins	Package	Macrocells	I/O	CLK/ Inputs	JTAG/V <sub>PP</sub>	V <sub>CC</sub> /GND
132	PQFP	160	92	4	5	31
196	PQFP	160	160	8	5	23

\*CHMOS is a patented process of Intel Corporation.

Electronic design has been a process of defining and implementing "black boxes". System level parameters are defined, and the system black box is broken into subsystem black boxes, which are subdivided again and again until the component level is reached. FPGAs were developed to function as large, highly-integrated black boxes to implement diverse logic functions, and the FLEXlogic FPGA family gives a designer the ultimate, flexible black box.

FLEXlogic FPGAs were designed to meet increasingly stringent design requirements. The first members of the FLEXlogic family can operate at 80 MHz system frequencies with predictable 10 ns pin-to-pin logic delays. FLEXlogic FPGAs are designed with Configurable

Function Blocks (CFB) that can function as 24V10-like logic or SRAM. The CFBs are interconnected with Intel's high-speed Global Interconnect Matrix that allows PLD-like performance in a high density device. Besides traditional sum-of-products and register logic functions, FLEXlogic CFBs can also perform fast identity compares or be configured as a block of 128 x 10 SRAM.

You can start developing with FLEXlogic now using Intel's free PLDshell Plus development tool. This tutorial will show you how to create a simple design using PLDshell Plus. You can also create FLEXlogic designs using the development tools that you now use; FLEXlogic FPGAs are supported on most third-party development tool systems.

2

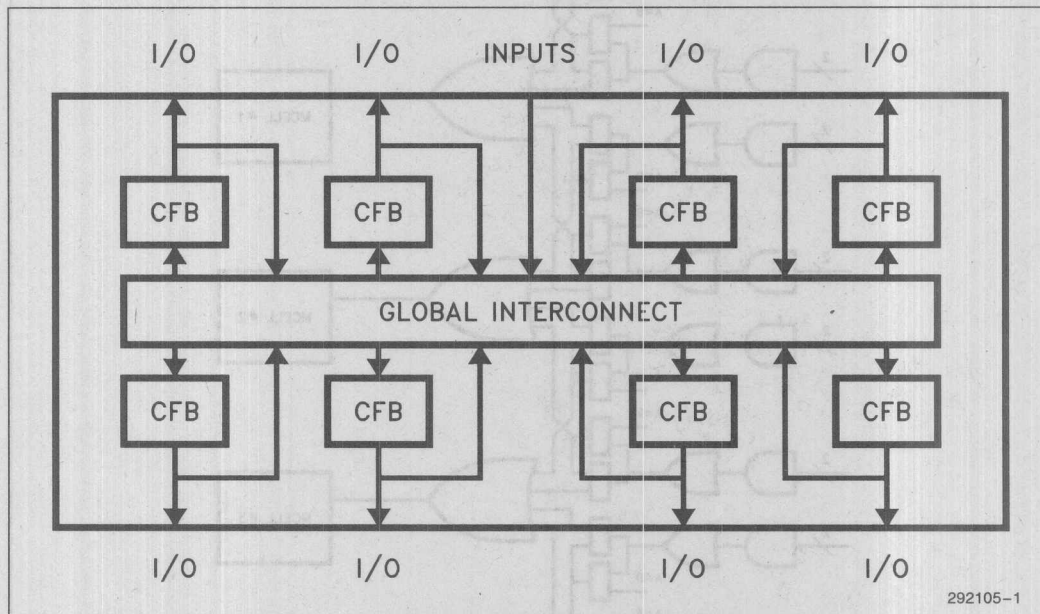


Figure 1. iFX780 Block Diagram

## Designing with FLEXlogic

FLEXlogic™ FPGAs are as easy to design with as the earliest PLDs; simply write the logical equations, develop a truth table, or enter the schematic equivalent.

$$\text{out1} = \text{in1} * \text{in3} \\ + \text{in4} * \text{in5} * \text{in6} * \text{in7} * \text{in8} * \text{in9} * \text{in10}$$

Up to 16 product terms can be included in a single sum-of-products equation. Most functions require three

or fewer product terms, but some functions require many more product terms to implement. Giving each macrocell enough resources to implement all functions is wasteful and expensive, but macrocells must also be able to implement large, complex functions. FLEXlogic™ uses an innovative product-term allocation scheme to maximize resource utilization and design fit. Pairs of product terms are steered from one macrocell to its neighbor, allowing macrocells to implement functions with up to 16 product terms.

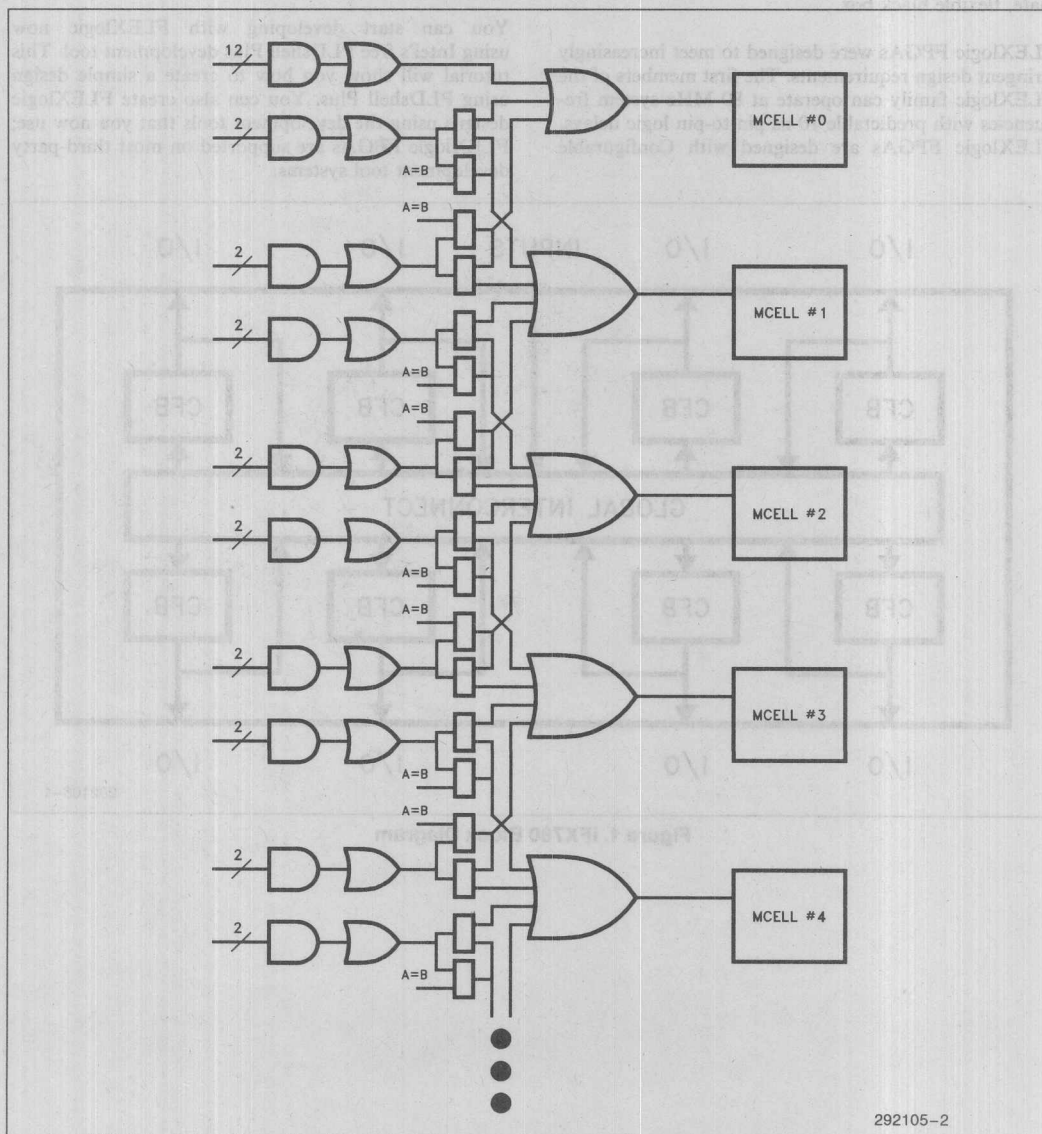


Figure 2. Product Term Allocation

## Identity Compare

Identity compares can be defined in parallel with other logic functions:

out2.CMP = [C[0:11]] == [D]0:11]]

The comparator uses the same inputs as other CFB logic, and works in parallel, so that compares can be included in logic equations, and still deliver the result in 10 ns.

requiring one pass through a Configurable Function Block take 10 ns. This includes 16 product-term equations and 12-bit identity compares.

Function results can be loaded into macrocell registers. Each register can be individually configured as a D or T register. SR and JK registers can also be emulated. Register clocking is user programmable in each macrocell, accommodating a variety of timing requirements. Registers can be clocked on the rising or falling edge of an external clock, a delayed external clock, or a function generated clock.

## Timing

Determining if FLEXlogic can meet your timing requirements is equally easy; all combinatorial functions

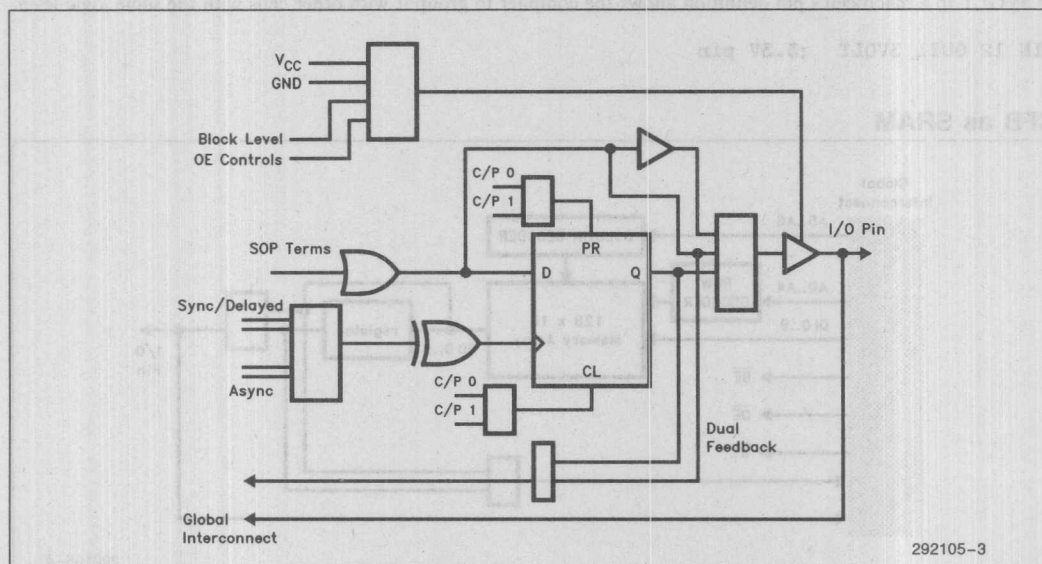


Figure 3. FLEXlogic Macrocell

```
PIN  clk1           ;define a synchronous clock
out1.CLKF = clk1     ;clock on rising edge
out2.CLKF = /clk1    ;clock on falling edge
out3.CLKF = clk1 DELAYCLK ;delayed clock
out4.CLKF = in8*in3*in4 ;function generated clock
```

The result, either registered or combinatorial, of each macrocell is always feedback to the Global Interconnect Matrix. The macrocell's I/O pin can be an output, input, or bi-directional, and is always available to the Global Interconnect Matrix.

```
out1.TRST = VCC      ;dedicated output
out2.TRST = GND      ;dedicated input
out3.TRST = in1*in2   ;bi-directional
```



## CFB as SRAM

Each CFB can be independently configured as 15 ns SRAM.

**PIN BUFFRAM[0:9] RAM**

BUFFRAM[0:6].ADDR = A0, A1, A2, A3, A4, A5, A6

BUFFRAM[0:9].DATA = DIN[0:9]

BUFFRAM.BE = in8

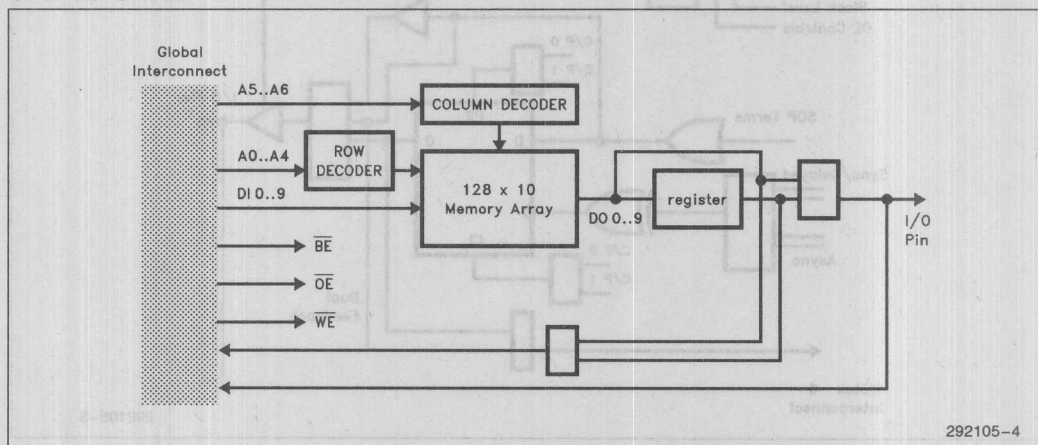
BUFFRAM.WE = write\_enable

## 3.3V/5V I/O

The physical limitations of silicon demand that high-performance electrical designs move to 3.3V or lower voltages. FLEXlogic FPGAs are the first programmable logic devices to address designers' needs for 3.3V and 5V logic. Each CFB can be configured as 3.3V or 5V logic by tying its V<sub>CC0</sub> pin to the appropriate supply voltage. Adding 3VOLT or 5VOLT to a macrocell's pin definition allows the compiler to group it with other cells with the same logic level.

PIN 12 OUT1 3VOLT ;3.3V pin

## CFB as SRAM



## Function, Pin, and JEDEC Compatible with EP600, EP610, EP610A, EP630, PALCE610, 85C060 and 5C060 PLDs

- t<sub>PD</sub> 10 ns, 100 MHz Counter Frequency (w/Internal Feedback)
- Clocking Speed Same as -7 ns PAL\* (74 MHz w/External Feedback)
- I<sub>CC</sub> = 105 mA max. @ 1 MHz
- 16 Macrocells with Programmable I/O Architecture (Register/Combinatorial). Registers Configurable as D/T/JK/RS Types
- Up to 20 Inputs (4 Dedicated and 16 I/O)
- 8 P-Terms, Selectable SOP Invert, Clear and OE P-Terms for Each Macrocell
- Programmable Clock System with 2 Synchronous Clocks and Asynchronous Clocking Option on all Registers
- Extensive Software and Programming Support via Intel and Third Party Tools
- 1-Micron CHMOS\* III-E EPROM Technology
- Programmable Low-Power Option for "Standby" Operation; 20  $\mu$ A Typ. in Standby Mode
- Programmable "Security Bit" Allows Total Protection of Proprietary Designs
- 100% Generically Tested Logic Array
- Available in 300-mil 24-Pin PDIP and 28 Pin PLCC Packages
- 85C060 also Available in 24-Pin CerDIP Package

(See Packaging Specifications Order Number #240800-001, Package Type N and P)

2

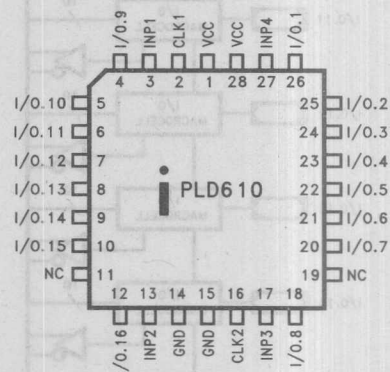
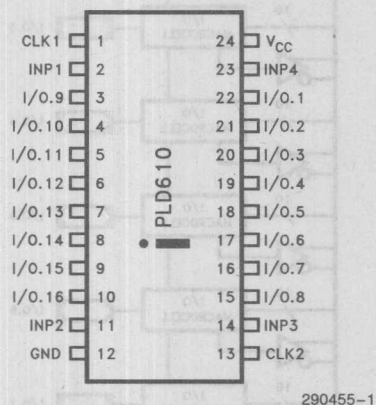


Figure 1. iPLD610 Pin Configurations

\*CHMOS is a patented process of Intel Corporation.

\*PAL is a registered trademark of Advanced Micro Devices, Inc.

# 85C060

All information in this document that refers to the iPLDxxx is identical to that of the 85C060.

## INTRODUCTION

The iPLD610 is a high-performance, high-integration, general-purpose CMOS PLD. The iPLD610 PLD (Programmable Logic Device) accommodates logic functions with up to 20 inputs and 16 I/O macrocells. Each I/O macrocell includes 8 product-terms (p-terms) for input, a separate clear p-term, and an output enable/asynchronous clock p-term. The iPLD610 is function-, pin-, and JEDEC-compatible with the EP600, EP610, EP610A, EP630, PALCE610, 85C060 and 5C060 PLDs. With a clock-

ing speed of 74 MHz (w/external feedback) the iPLD610 offers a higher integration, lower power alternative to registered -7 ns PALs/GALs.

The iPLD610 uses CHMOS EPROM (floating gate) cells as logic control elements instead of fuses. The CHMOS EPROM technology reduces power consumption in comparison to bipolar devices without sacrificing speed performance. In addition, Intel's advanced CHMOS III-E EPROM process technology enables higher logic densities to be achieved with superior speed and low-power performance over other comparable devices. Intel's  $\mu$ PLDs add the benefits of "zero" stand-by power not available on other programmable logic devices. EPROM technology allows these devices to be 100% factory tested by programming and erasing all the EPROM logic control elements.

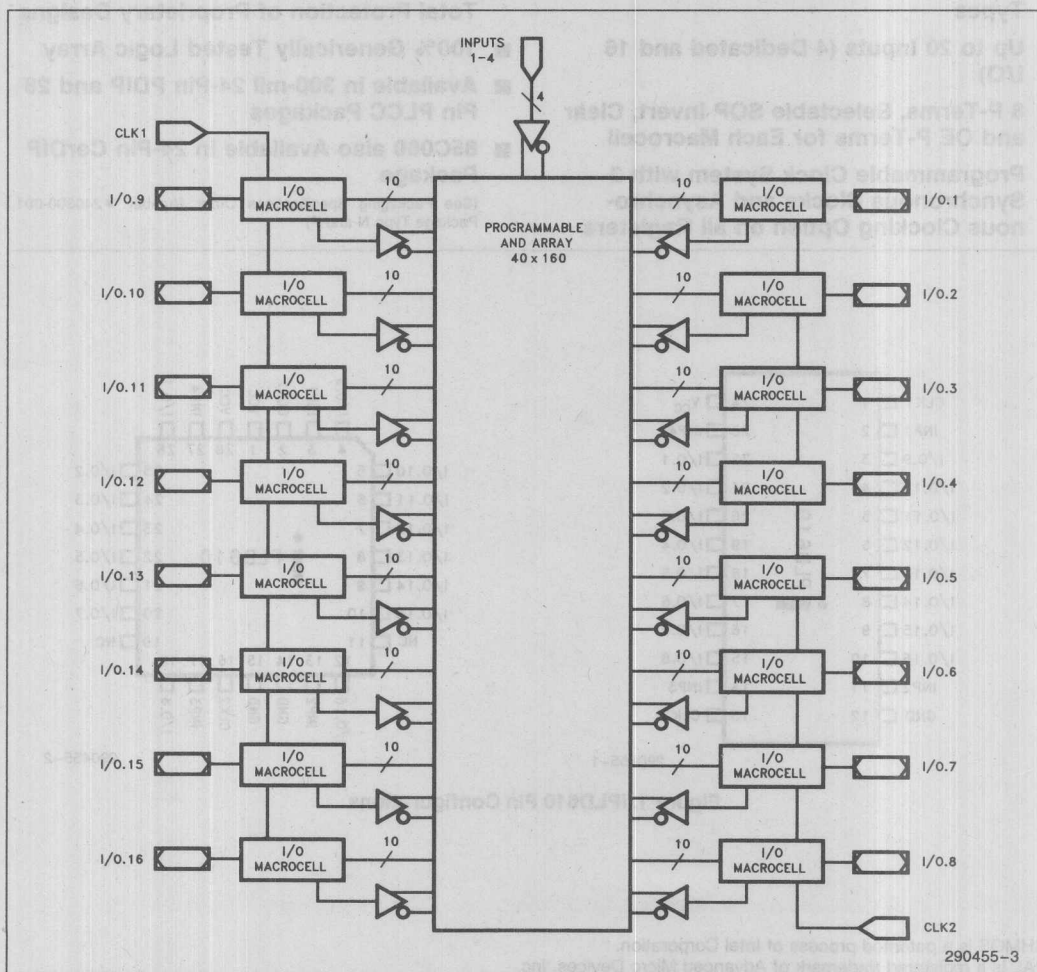


Figure 2. iPLD610 Global Architecture

The architecture of the iPLD610 is based on the "Sum of Products" PLA (Programmable Logic Array) structure with a programmable AND array feeding into a fixed OR array. The device accommodates combinational and sequential logic functions. A programmable I/O architecture provides individual selection of either combinational or registered output and feedback signals all with selectable polarity.

A feature unique to the iPLD610 is the ability to individually program the output registers as a D-, T-, SR-, or JK-type Flip-Flop without sacrificing the utilization of programmable AND logic. Each output register can be individually clocked from any of the input or feedback paths available within the AND array. With these features, a wide variety of logic functions can be simultaneously implemented—all on the same device.

## ARCHITECTURE DESCRIPTION

Externally, the iPLD610 has 4 dedicated data input pins, 16 I/O pins that may be configured for input, output, or bidirectional operations, and 2 synchronous clock inputs. The iPLD610 is contained in a 24-pin ceramic windowed or OTP plastic (300 mils) or 28-lead OTP J-leaded chip carrier package.

The basic Macrocell architecture for the iPLD610 is shown in Figure 3. The iPLD610 has 16 of these Macrocells (one for each I/O pin). The Macrocell is organized in the familiar sum-of-products structure with a programmable AND array attached to a fixed OR term. The inputs to the programmable AND array originate from the true and complement signals from each of the dedicated input pins and each of the I/O control blocks. The 40-input AND array of the iPLD610 feeds 160 AND gates (product terms) which are distributed among the 16 Macrocells in the device.

2

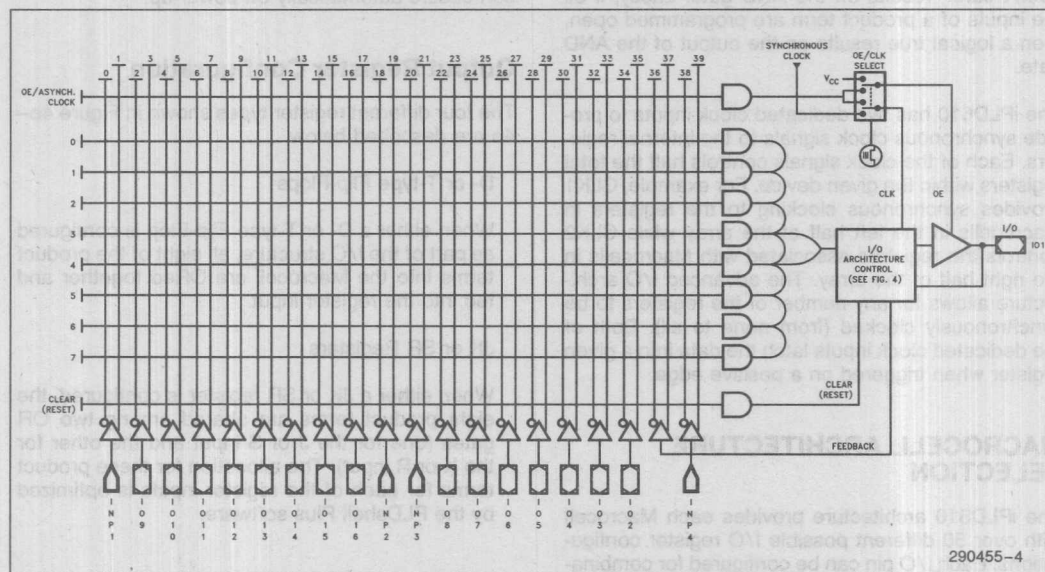


Figure 3. iPLD610 Macrocell Architecture



Each Macrocell contains ten product terms. Eight of the ten product terms (AND gates) are dedicated for the SOP logic implementation. One product term on each Macrocell is used for RESET control to the output register associated with the Macrocell. The final product term is used for OE/Asynchronous Clock implementation.

Within the AND array, there is an EPROM connection at every intersection of an input signal (true and complement) and a product term to a given Macrocell. Before programming an erased device, every EPROM connection is made at every intersection. But during the programming process, these connections are opened so that only the desired connections remain. Therefore, the true or complement of any input signal can be connected to any product term. If both the true and complement connections of any signal are left intact, a logical false results on the output of the AND gate. However, if both the true and complement connections are open, then a logic "don't care" results on the AND gate. Lastly, if all the inputs of a product term are programmed open, then a logical true results on the output of the AND gate.

The iPLD610 has two dedicated clock inputs to provide synchronous clock signals to the internal registers. Each of the clock signals controls half the total registers within the given device. For example, CLK1 provides synchronous clocking to the registers in Macrocells in the left half of the array while CLK2 controls the registers associated with Macrocells in the right half of the array. The advanced I/O architecture allows for any number of the registers to be synchronously clocked (from none to all). Both of the dedicated clock inputs latch the data into a given register when triggered on a positive edge.

## MACROCELL ARCHITECTURE SELECTION

The iPLD610 architecture provides each Macrocell with over 50 different possible I/O register configurations. Each I/O pin can be configured for combinatorial or registered output (true or complement) with feedback. In addition, four different types of output registers can be implemented on I/O pin without any additional logic requirements. The feedback mechanism for each register back into the AND array can be programmed to provide for either registered feedback from the Macrocell or input feedback (treating the pin as an input). Another advantage of the advanced I/O capability of the iPLD610 is the ability to individually clock each internal register from asynchronous clock signals.

## Invert Select EPROM Bit

The Invert Select EPROM bit is used to invert the product term input into the register. This applies to all inputs including double inputs on the JK and SR registers.

## REGISTER SELECTION

The advanced I/O architecture of the iPLD610 allows four different register types along with combinatorial output as illustrated in Figure 4a. The register types include a T, D, JK, or SR Flip-Flop and each Macrocell I/O structure may be independently configured. In addition, all registers have an individual asynchronous RESET control from a dedicated product term derived in the AND array. When this dedicated product term is a logical one, the Macrocell register is immediately cleared to a logical zero independent of the register clock. The RESET function occurs automatically on power-up.

## Output Register Configuration

The four different register types shown in Figure 4b-4e are described below.

### D- or T-type Flip-Flops

When either a D- or T-type Flip-Flop is configured as part of the I/O structure, all eight of the product terms into the Macrocell are ORed together and fed into the register input.

### JK or SR Registers

When either a JK or SR register is configured, the eight product terms are shared among two OR gates (one for the J or S input and the other for the K or R input). The allocation for these product terms for each of the register inputs is optimized by the PLDshell Plus software.

## OUTPUT/FEEDBACK

The Output Select Multiplexer allows for either registered, combinatorial or no output.

The Feedback Select Multiplexer EPROM bit enables registered, I/O (using the pin for bidirectional input or just input), or no feedback to the AND array.

The Feedback Select is also important for building equations with more than 8 product terms. The 8-product product term of a Macrocell can be fed back into the AND array and combined with still more signals to create a much larger product term (of more than 8-inputs). If the feedback term is not to be used as an output, the associated Macrocell pin

should be left floating (no connect) when assembled onto a circuit board.

Any I/O pin may be configured as a dedicated input by selecting no output and pin feedback through the appropriate multiplexers.

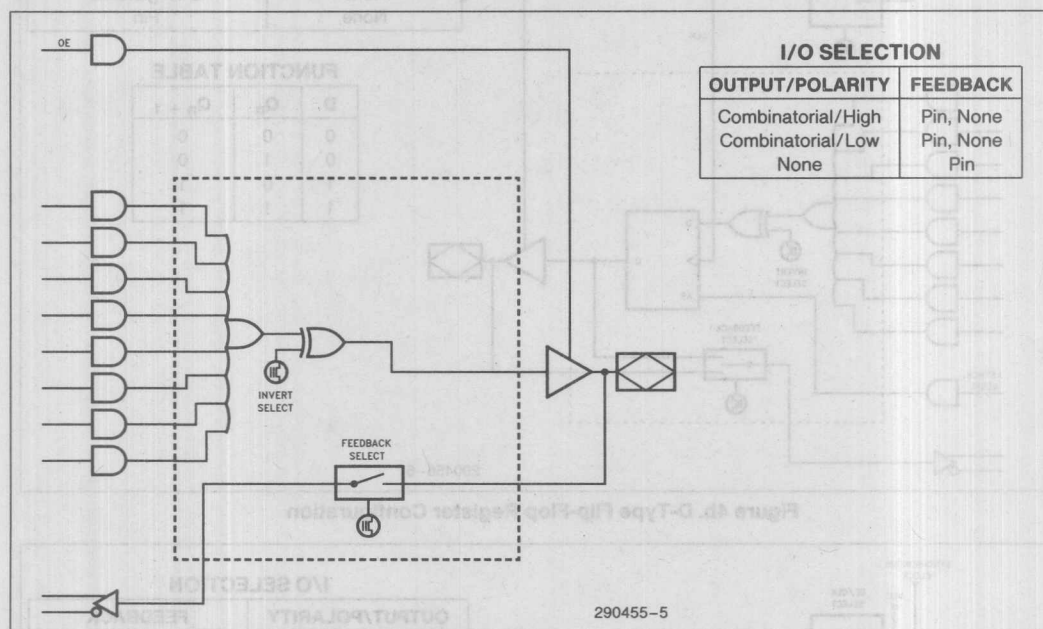


Figure 4a. Combinatorial I/O Configuration

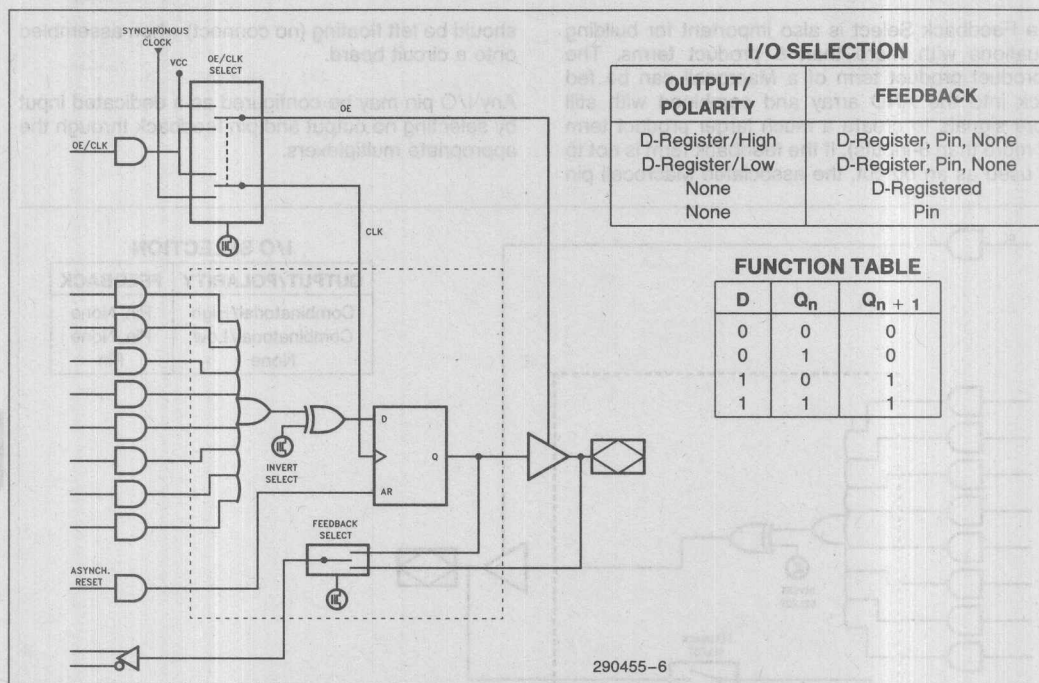


Figure 4b. D-Type Flip-Flop Register Configuration

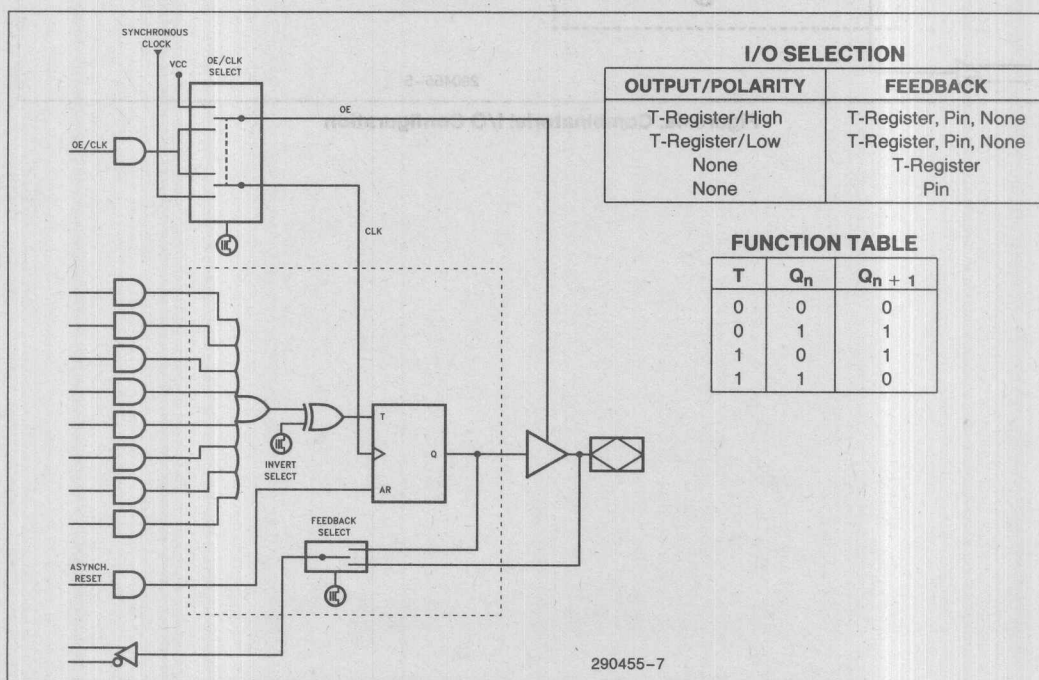


Figure 4c. Toggle Flip-Flop Register Configuration

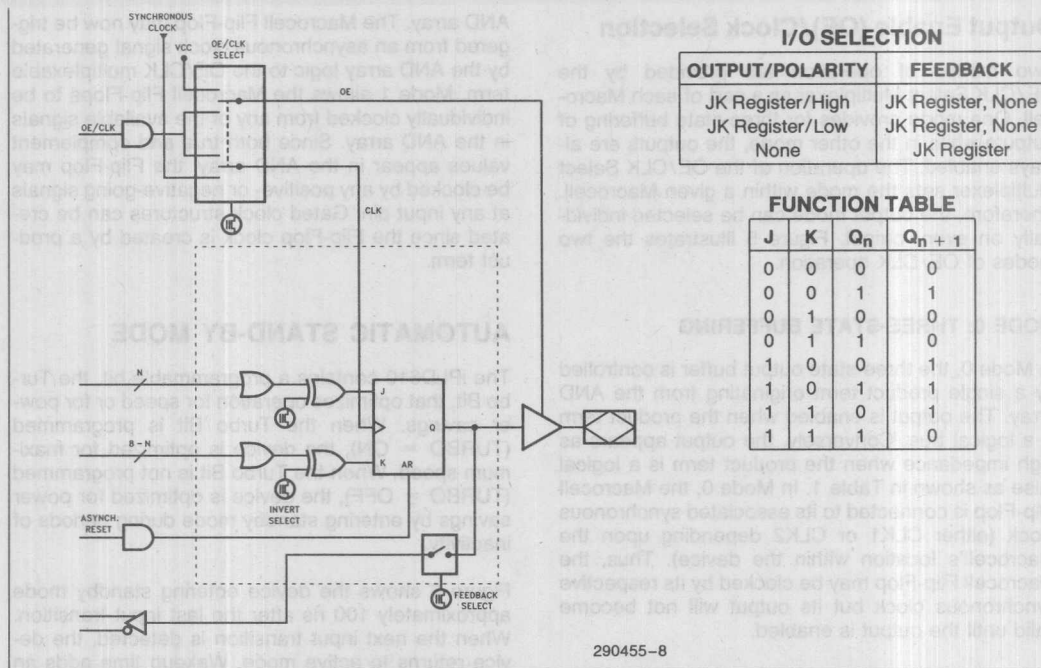


Figure 4d. JK Flip-Flop Register Configuration

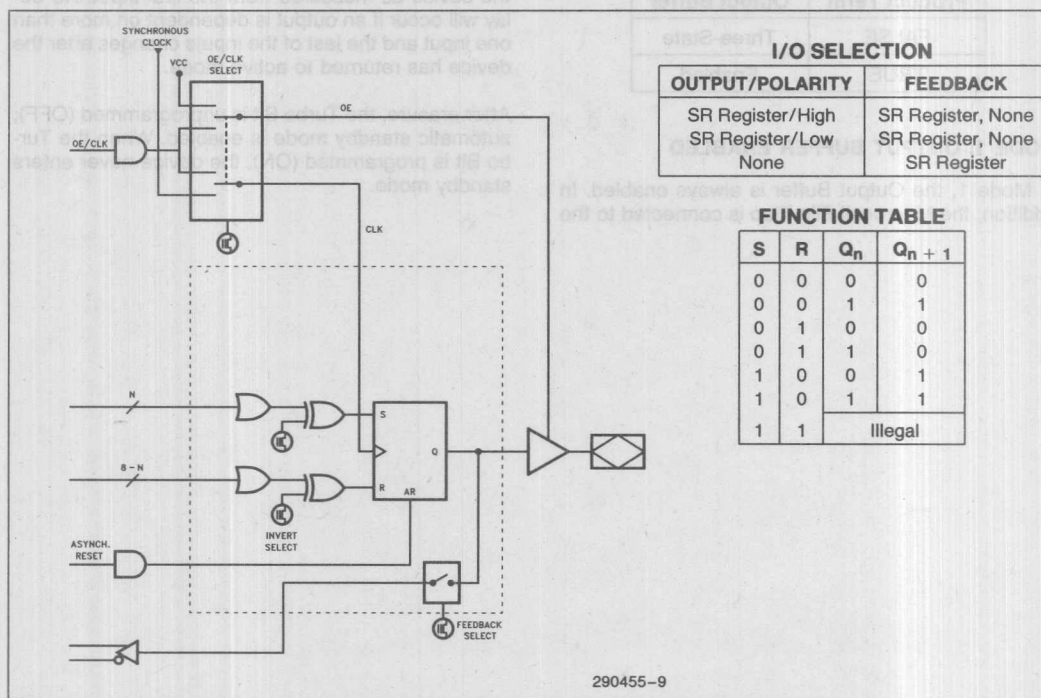


Figure 4e. SR Flip-Flop Register Configuration



## Output Enable (OE)/Clock Selection

Two modes of operation are provided by the OE/CLK Select Multiplexer as a part of each Macrocell. One mode provides for three-state buffering of outputs while in the other mode, the outputs are always enabled. The operation of the OE/CLK Select Multiplexer sets the mode within a given Macrocell. Therefore, the output mode can be selected individually on every output. Figure 5 illustrates the two modes of OE/CLK operation.

### MODE 0: THREE-STATE BUFFERING

In Mode 0, the three-state output buffer is controlled by a single product term originating from the AND array. The output is enabled when the product term is a logical true. Conversely, the output appears as high impedance when the product term is a logical false as shown in Table 1. In Mode 0, the Macrocell Flip-Flop is connected to its associated synchronous clock (either CLK1 or CLK2 depending upon the Macrocell's location within the device). Thus, the Macrocell Flip-Flop may be clocked by its respective synchronous clock but its output will not become valid until the output is enabled.

Table 1. Mode 0 Output Selection

Product Term	Output Buffer
FALSE	Three-State
TRUE	Enabled

### MODE 1: OUTPUT BUFFER ENABLED

In Mode 1, the Output Buffer is always enabled. In addition, the Macrocell Flip-Flop is connected to the

AND array. The Macrocell Flip-Flop may now be triggered from an asynchronous clock signal generated by the AND array logic to the OE/CLK multiplexable term. Mode 1 allows the Macrocell Flip-Flops to be individually clocked from any of the available signals in the AND array. Since both true and complement values appear in the AND array, the Flip-Flop may be clocked by any positive- or negative-going signals at any input pin. Gated clock structures can be created since the Flip-Flop clock is created by a product term.

### AUTOMATIC STAND-BY MODE

The iPLD610 contains a programmable bit, the Turbo Bit, that optimizes operation for speed or for power savings. When the Turbo Bit is programmed (TURBO = ON), the device is optimized for maximum speed. When the Turbo Bit is not programmed (TURBO = OFF), the device is optimized for power savings by entering standby mode during periods of inactivity.

Figure 6 shows the device entering standby mode approximately 100 ns after the last input transition. When the next input transition is detected, the device returns to active mode. Wakeup time adds an additional 25 ns to the propagation delay through the device as measured from the first input. No delay will occur if an output is dependent on more than one input and the last of the inputs changes after the device has returned to active mode.

After erasure, the Turbo Bit is unprogrammed (OFF); automatic standby mode is enabled. When the Turbo Bit is programmed (ON), the device never enters standby mode.

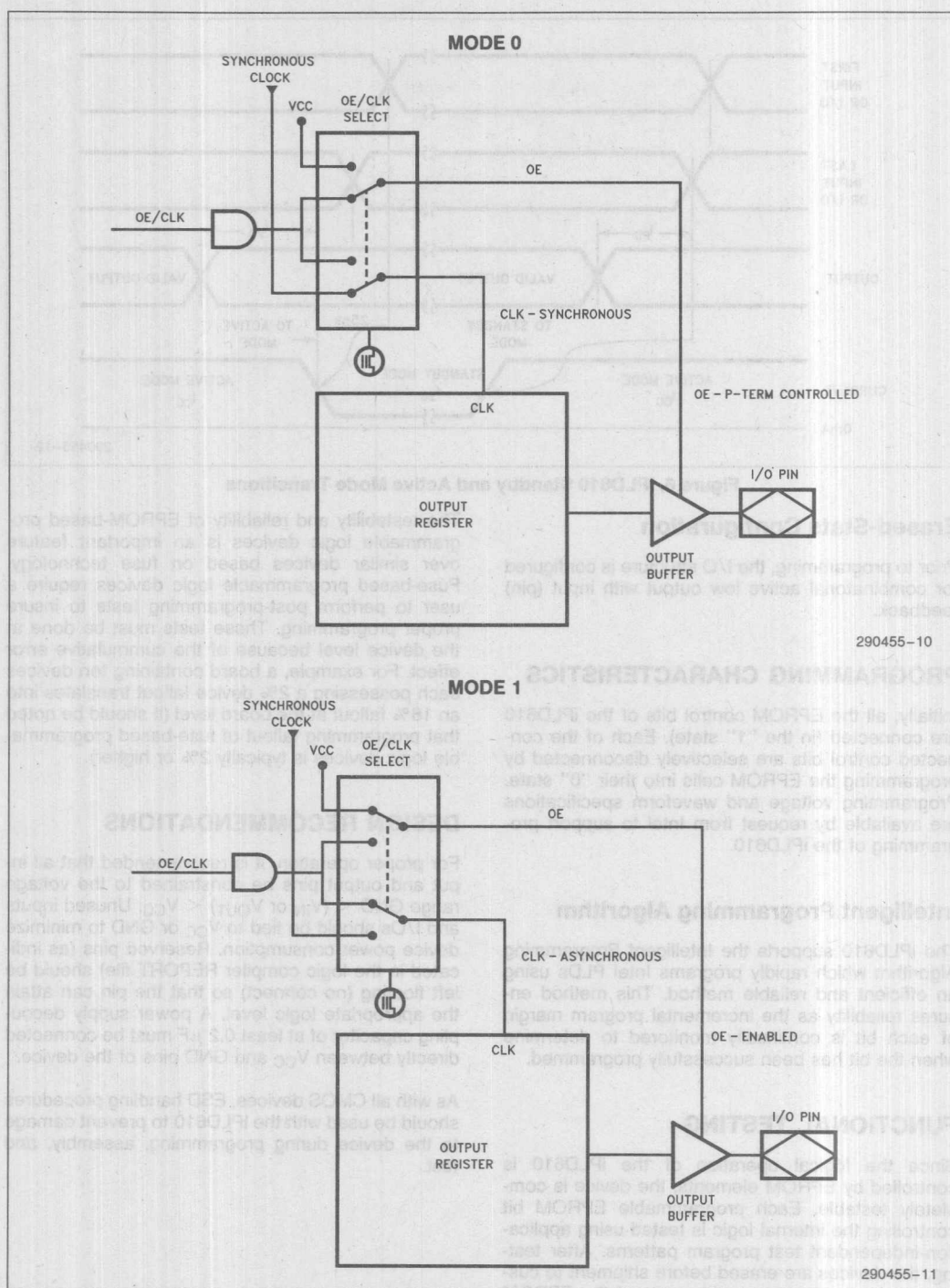


Figure 5. Output Enable/Clock Configuration

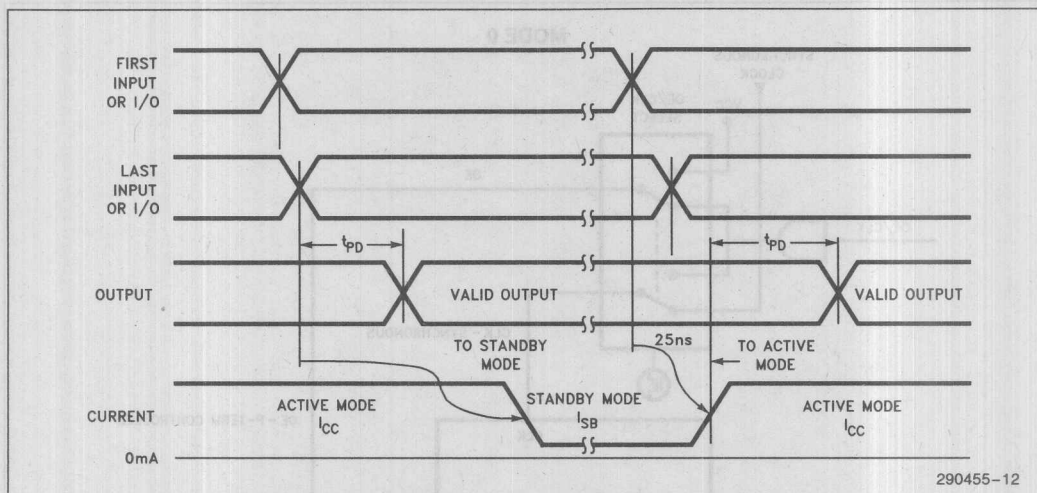


Figure 6. iPLD610 Standby and Active Mode Transitions

### Erased-State Configuration

Prior to programming, the I/O structure is configured for combinatorial active low output with input (pin) feedback.

### PROGRAMMING CHARACTERISTICS

Initially, all the EPROM control bits of the iPLD610 are connected (in the "1" state). Each of the connected control bits are selectively disconnected by programming the EPROM cells into their "0" state. Programming voltage and waveform specifications are available by request from Intel to support programming of the iPLD610.

### Intelligent Programming Algorithm

The iPLD610 supports the Intelligent Programming Algorithm which rapidly programs Intel PLDs using an efficient and reliable method. This method ensures reliability as the incremental program margin of each bit is continually monitored to determine when the bit has been successfully programmed.

### FUNCTIONAL TESTING

Since the logical operation of the iPLD610 is controlled by EPROM elements, the device is completely testable. Each programmable EPROM bit controlling the internal logic is tested using application-independent test program patterns. After testing, the devices are erased before shipment to customers. No post-programming tests of the EPROM array are required.

The testability and reliability of EPROM-based programmable logic devices is an important feature over similar devices based on fuse technology. Fuse-based programmable logic devices require a user to perform post-programming tests to insure proper programming. These tests must be done at the device level because of the cumulative error effect. For example, a board containing ten devices each possessing a 2% device fallout translates into an 18% fallout at the board level (it should be noted that programming fallout of fuse-based programmable logic devices is typically 2% or higher).

### DESIGN RECOMMENDATIONS

For proper operation, it is recommended that all input and output pins be constrained to the voltage range  $GND < (V_{IN} \text{ or } V_{OUT}) < V_{CC}$ . Unused inputs and I/Os should be tied to  $V_{CC}$  or  $GND$  to minimize device power consumption. Reserved pins (as indicated in the logic compiler REPORT file) should be left floating (no connect) so that the pin can attain the appropriate logic level. A power supply decoupling capacitor of at least  $0.2 \mu F$  must be connected directly between  $V_{CC}$  and  $GND$  pins of the device.

As with all CMOS devices, ESD handling procedures should be used with the iPLD610 to prevent damage to the device during programming, assembly, and test.

## DESIGN SECURITY

A single EPROM bit provides a programmable design security feature that controls the access to the data programmed into the device. If this bit is set, a proprietary design within the device cannot be copied. This EPROM security bit enables a higher degree of design security than fused-based devices since programmed data within EPROM cells is invisible even to microscopic evaluation. The EPROM security bit, along with all the other EPROM control bits, will be reset by erasing the device.

## LATCH-UP IMMUNITY

All of the input, I/O, and clock pins of the iPLD610 have been designed to resist latch-up which is inherent in inferior CMOS structures. The iPLD610 is designed with Intel's proprietary CHMOS II-E EPROM process. Thus, each of the pins will not experience latch-up with currents up to  $\pm 100$  mA and voltages ranging from  $-1V$  to  $(V_{CC} + 1V)$ . Furthermore, the programming pin is designed to resist latch-up to the 13.5V maximum device limit.

## DEVELOPMENT SOFTWARE

### Third Party Support

The iPLD610 is supported by third-party logic compilers such as ABEL\*, CUPL\*, PLDesigner\*, Log/IC,

etc. Programming support is provided by third-party programmer companies such as Data I/O, Logical Devices, STAG, etc. Please refer to the "Third-Party Support" lists in the *Programmable Logic* handbook for complete information and vendor contacts.

### PLDshell Plus

PLDshell Plus design software is Intel's new, user-friendly design tool for PLD design. PLDshell Plus allows user's to incorporate their preferred text editor, programming software, and additional design tools into a easy-to-use, menued design environment that includes Intel's PLDasm™ logic compiler and simulation software along with disassembly, conversion, and translation utilities. The PLDasm compiler and simulator software accepts industry-standard PDS source files that express designs as Boolean equations, truth tables, or state machines. On-line help, datasheet briefs, technical notes, and error message information, along with waveform viewing/printing capability make the design task as easy as possible. PLDshell Plus software is available from Intel Literature channels or from your local Intel sales representative. Order #468810.

Tools that support schematic capture and timing simulation for the iPLD610 are available. Support under iPLS II is still available. Please refer to the "Development Tools" section of the *Programmable Logic* handbook.

Commercial	PDP	iPLD610-10	10	60	20
Commercial	PLCC	iPLD610-15	15	60	20
Commercial	*CDIP	iPLD610-15	15	60	20
Commercial	PDP	iPLD610-15	15	60	20
Commercial	PLCC	iPLD610-15	15	60	20
Industrial	*CDIP	iPLD610-15	15	60	20
Industrial	PLCC	iPLD610-15	15	60	20
Commercial	PDP	iPLD610-25	25	60	40
Commercial	PLCC	iPLD610-25	25	60	40
Commercial	*CDIP	iPLD610-25	25	60	40
Commercial	PDP	iPLD610-25	25	60	40
Commercial	PLCC	iPLD610-25	25	60	40
Industrial	*CDIP	iPLD610-25	25	60	40
Industrial	PLCC	iPLD610-25	25	60	40

\*ABEL is a trademark of Data I/O Corp. CUPL is a trademark of Logical Devices, Inc. PLDesigner is a trademark of MINC, Inc. Log/IC is a trademark of ISDATA, Corporation.



## ADF PRIMITIVES SUPPORTED

The following ADF primitives are supported by this device:

INP	JOJF
CONF	JONF
COIF	SONF
RONF	SOSF
RORF	TOIF
ROIF	TONF
NORF	TOTF
NOJF	CLKB
NOSF	
NOTF	

## ORDERING INFORMATION

f <sub>CNT1</sub> (MHz)	f <sub>CNT2</sub> (MHz)	t <sub>PD</sub> (ns)	Order Code	Package	Operating Range
74	100	10	P PLD610-10	PDIP	Commercial
			N PLD610-10	PLCC	Commercial
			D85C060-10	*CerDIP	Commercial
			P85C060-10	PDIP	Commercial
			N85C060-10	PLCC	Commercial
66	83.3	12	P PLD610-12	PDIP	Commercial
			N PLD610-12	PLCC	Commercial
			P85C060-12	PDIP	Commercial
			N85C060-12	PLCC	Commercial
50	66	15	P PLD610-15	PDIP	Commercial
			N PLD610-15	PLCC	Commercial
			D85C060-15	*CerDIP	Commercial
			P85C060-15	PDIP	Commercial
			N85C060-15	PLCC	Commercial
			TD85C060-15	*CerDIP	Industrial
40	40	25	TN85C060-15	PLCC	Industrial
			P PLD610-25	PDIP	Commercial
			N PLD610-25	PLCC	Commercial
			D85C060-25	*CerDIP	Commercial
			P85C060-25	PDIP	Commercial
			N85C060-25	PLCC	Commercial
			TD85C060-25	*CerDIP	Industrial
			TN85C060-25	PLCC	Industrial

\*Windowed CerDIP package allows UV erase.

**ABSOLUTE MAXIMUM RATINGS\***

Symbol	Parameter	Min	Max	Units
V <sub>CC</sub>	Supply Voltage(1)	-2.0	7.0	V
V <sub>PP</sub>	Programming Supply Voltage(1)	-2.0	13.5	V
V <sub>I</sub>	DC Input Voltage(1)(2)	-0.5	V <sub>CC</sub> + 0.5	V
t <sub>stg</sub>	Storage Temperature	-65	+150	°C
t <sub>amb</sub>	Ambient Temperature(3)	-10	+85	°C

**NOTES:**

1. Voltages with respect to ground.
2. Minimum DC input is -0.5V. During transitions, the inputs may undershoot to -2.0V or overshoot to +7.0V for periods less than 20 ns under no load conditions.
3. Under bias. Extended temperature versions are also available.

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

*\*WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

**RECOMMENDED OPERATING CONDITIONS**

Symbol	Parameter	Min	Max	Unit
V <sub>CC</sub>	Supply Voltage	4.75	5.25	V
V <sub>IN</sub>	Input Voltage	0	V <sub>CC</sub>	V
V <sub>O</sub>	Output Voltage	0	V <sub>CC</sub>	V
T <sub>A</sub>	Operating Temperature	0	+70	°C
t <sub>R</sub>	Input Rise Time		500	ns
t <sub>F</sub>	Input Fall Time		500	ns

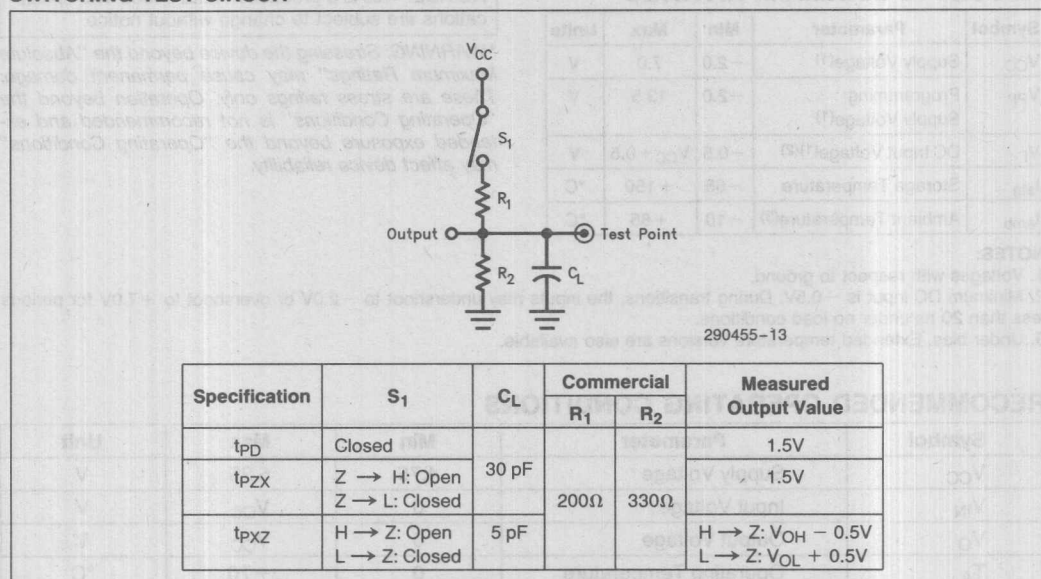
**D.C. CHARACTERISTICS** T<sub>A</sub> = 0°C to 70°C, V<sub>CC</sub> = 5.0V ± 5%

Symbol	Parameter	Min	Typ	Max	Unit	Conditions
V <sub>IH</sub> (4)	HIGH Level Input Voltage	2.0		V <sub>CC</sub> + 0.3	V	
V <sub>IL</sub> (4)	LOW Level Input Voltage	-0.3		0.8	V	
V <sub>OH</sub>	HIGH Level Output Voltage	2.4			V	I <sub>O</sub> = -4.0 mA DC, V <sub>CC</sub> = Min.
V <sub>OL</sub> (5)	LOW Level Output Voltage			0.45	V	I <sub>O</sub> = 12.0 mA DC, V <sub>CC</sub> = Min.
I <sub>I</sub>	Input Leakage Current	-10		10	μA	V <sub>CC</sub> = Max., GND < V <sub>IN</sub> < V <sub>CC</sub>
I <sub>OZ</sub>	Output Leakage Current	-10		10	μA	V <sub>CC</sub> = Max., GND < V <sub>OUT</sub> < V <sub>CC</sub>
I <sub>SC</sub> (6)	Output Short Circuit Current	-30		-120	mA	V <sub>CC</sub> = Max., V <sub>OUT</sub> = 0.5V
I <sub>SB</sub> (7)	Standby Current		20	150	μA	V <sub>CC</sub> = Max., V <sub>IN</sub> = V <sub>CC</sub> or GND, Standby Mode
I <sub>CC</sub>	Power Supply Current (See I <sub>CC</sub> vs. Freq. Graph)		3	8	mA	V <sub>CC</sub> = Max, V <sub>IN</sub> = V <sub>CC</sub> or GND, No Load, f <sub>IN</sub> = 1 MHz, Device Prog. as 16-Bit Counter, Turbo = Off
			65	105	mA	Turbo = On, f <sub>IN</sub> = 1 MHz
I <sub>CCI</sub>	Industrial Temperature Power Supply Current			150	mA	Turbo = On, f <sub>IN</sub> = 1 MHz

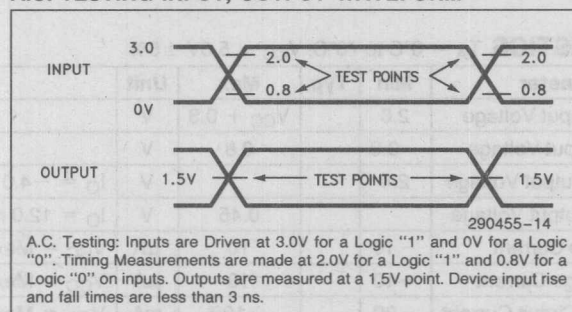
**NOTES:**

4. Absolute values with respect to device GND; all over and undershoots due to system or tester noise are included.
5. Maximum DC I<sub>OL</sub> for the device is 64 mA for CLK1 group I/O. 1-I/O.8 and 64 mA for CLK2 group I/O.9-I/O16.
6. Not more than 1 output should be tested at a time. Duration of that test must not exceed 1 second.
7. In Non-Turbo Mode (TURBO = OFF), device enters standby mode approximately 75 ns after the last input transition. I<sub>SB</sub> is measured with the window covered (CerDIP).

# SWITCHING TEST CIRCUIT



## A.C. TESTING INPUT, OUTPUT WAVEFORM



## CAPACITANCE (T<sub>A</sub> = 0°C to +70°C; V<sub>CC</sub> = 5.0V ± 5%)(8)

Symbol	Parameter	Min	Typ	Max	Unit	Conditions
C <sub>IN</sub>	Input Capacitance		5	8	pF	V <sub>IN</sub> = 0V, f = 1.0 MHz
C <sub>IO</sub>	I/O Capacitance		6	8	pF	V <sub>OUT</sub> = 0V, f = 1.0 MHz
C <sub>CLK</sub>	CLK Capacitance		8	10	pF	V <sub>IN</sub> = 0V, f = 1.0 MHz
C <sub>VPP</sub>	V <sub>PP</sub> Pin Capacitance		10	12	pF	V <sub>PP</sub> on CLK2, f = 1.0 MHz

### NOTE:

8. These values are evaluated during initial characterization and whenever design modifications occur that may affect capacitance.

**COMBINATORIAL MODE A.C. CHARACTERISTICS**(T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5.0V ± 5%)(9)

Symbol	Parameter	iPLD610-10			iPLD610-15			iPLD610-25			Non-Turbo <sup>(10)</sup> Mode	Units
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
t <sub>PD1</sub> <sup>(11)</sup>	Input to Output Valid			10			15			25	+ 25	ns
t <sub>PD2</sub> <sup>(11)</sup>	I/O to Output Valid			10			15			25	+ 25	ns
t <sub>PZX</sub> <sup>(12)</sup>	Input or I/O to Output Enable			15			18			25	+ 25	ns
t <sub>PXZ</sub> <sup>(12)</sup>	Input or I/O to Output Disable			13			18			25	+ 25	ns
t <sub>CLR</sub>	Input or I/O to Async. Reset			13			18			25	+ 25	ns

**NOTES:**9. Typical values are at T<sub>A</sub> = 25°C, V<sub>CC</sub> = 5V, Active Mode.

10. If device is operated in Non-Turbo Mode (TURBO = OFF), and the device is inactive for approx. 75 ns, increase time by amount shown.

11. Measured with eight outputs switching. See t<sub>PD</sub> vs. Number of Outputs Switching graph.12. t<sub>PZX</sub> and t<sub>PXZ</sub> are measured at ±0.5V from steady state voltage as driven by spec. output load.

13. Measured with device configured as a 16-bit counter.

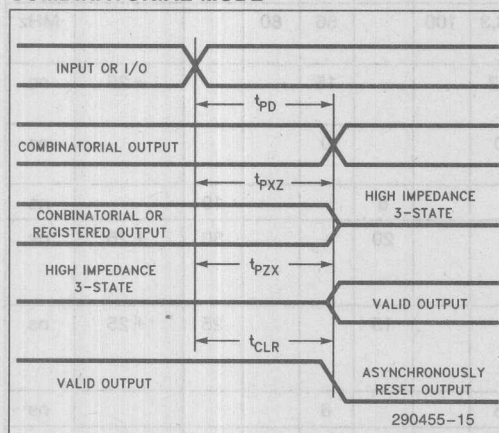
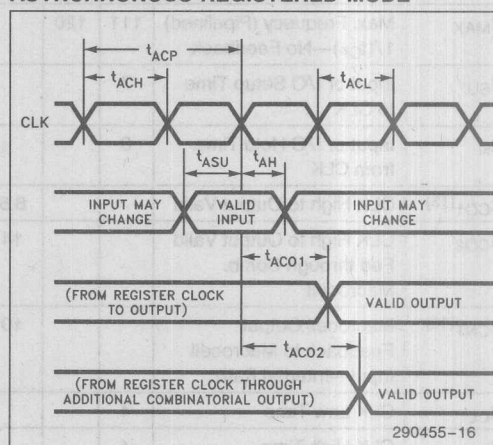
**REGISTER MODE—SYNCHRONOUS CLOCK A.C. CHARACTERISTICS**T<sub>A</sub> = 0°C to 70°C, V<sub>CC</sub> = 5.0V ± 5%<sup>(9)</sup>

Symbol	Parameter	iPLD610-10			iPLD610-15			iPLD610-25			Non-Turbo <sup>(10)</sup> Mode	Units
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
f <sub>CNT1</sub> <sup>(13)</sup>	Max. Counter Frequency 1/(t <sub>SU</sub> + t <sub>CO1</sub> )—Ext. Feedback	74	85		50	66		40	50			MHz
f <sub>CNT2</sub> <sup>(13)</sup>	Max. Counter Frequency 1/(t <sub>CNT</sub> )—Internal Feedback	100	111		66	75		40	66			MHz
f <sub>MAX</sub>	Max. Frequency (Pipelined) 1/(t <sub>CP</sub> )—No Feedback	111	120		83.3	100		66	80			MHz
t <sub>SU</sub>	Input or I/O Setup Time to CLK	7			12			15			+ 25	ns
t <sub>H</sub>	Input or I/O Hold Time from CLK	0			0			0				ns
t <sub>CO1</sub> <sup>(13)</sup>	CLK High to Output Valid			6.5			8			10		ns
t <sub>CO2</sub>	CLK High to Output Valid Fed through Comb. Macrocell			14			20			30	+ 25	ns
t <sub>CNT</sub> <sup>(13)</sup>	Macrocell Output Feedback to Macrocell Input—Internal Path			10			15			25	+ 25	ns
t <sub>CL</sub>	CLK Low Time	4			5			6				ns
t <sub>CH</sub>	CLK High Time	4			5			6				ns
t <sub>CP</sub>	CLK Period	9			12			15				ns

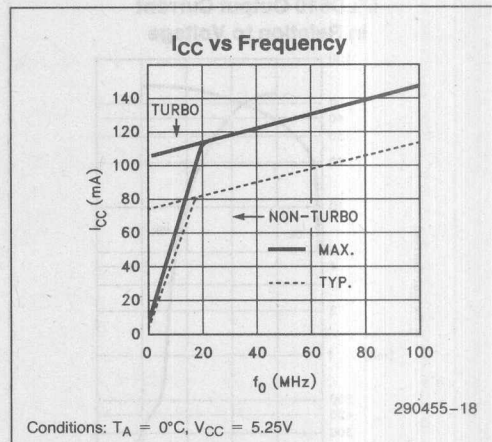
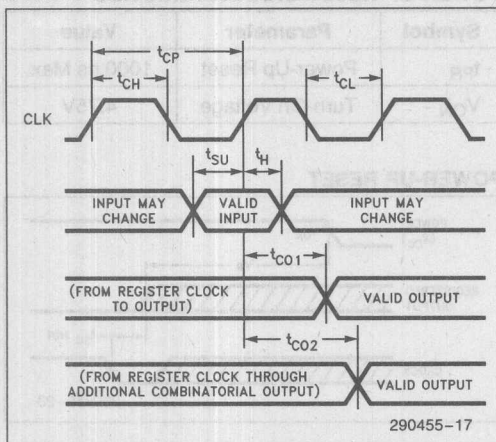


**REGISTER MODE—ASYNCHRONOUS CLOCK A.C. CHARACTERISTICS** $T_A = 0^\circ\text{C}$  to  $70^\circ\text{C}$ ,  $V_{CC} = 5.0\text{V} \pm 5\%$ <sup>(9)</sup>

Symbol	Parameter	iPLD610-10			iPLD610-15			iPLD610-25			Non-Turbo <sup>(10)</sup> Mode	Units
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
$f_{ACNT1}^{(13)}$	Max. Counter Frequency $1/(t_{ASU} + t_{ACO1})$ —Ext. Feedback	71.4	80		50	66		33.3	40			MHz
$f_{ACNT2}^{(13)}$	Max. Counter Frequency $1/t_{ACNT}$ —Internal Feedback	100	111		66	75		40	50			MHz
$f_{AMAX}$	Max. Frequency (Pipelined) $1/t_{ACP}$ —No Feedback	100	111		66	75		50	66			MHz
$t_{ASU}$	Input or I/O Setup Time to Asynch. CLK	2			4			5			+25	ns
$t_{AH}$	Input or I/O Hold Time from Asynch. CLK	3			6			8				ns
$t_{ACO1}^{(13)}$	Asynch. CLK High to Output Valid			12			16			25	+25	ns
$t_{ACO2}$	Asynch. CLK High to Output Valid Fed through Comb. Macrocell			20			30			45	+25	ns
$t_{ACNT}$	Macrocell Output Feedback to Macrocell Input—Internal Path			10			15			25	+25	ns
$t_{ACL}$	Asynch. CLK Low Time	4			6			7				ns
$t_{ACH}$	Asynch. CLK High Time	4			6			7				ns
$t_{ACP}$	Asynch. CLK Period	10			15			20				ns

**COMBINATORIAL MODE****ASYNCHRONOUS REGISTERED MODE**

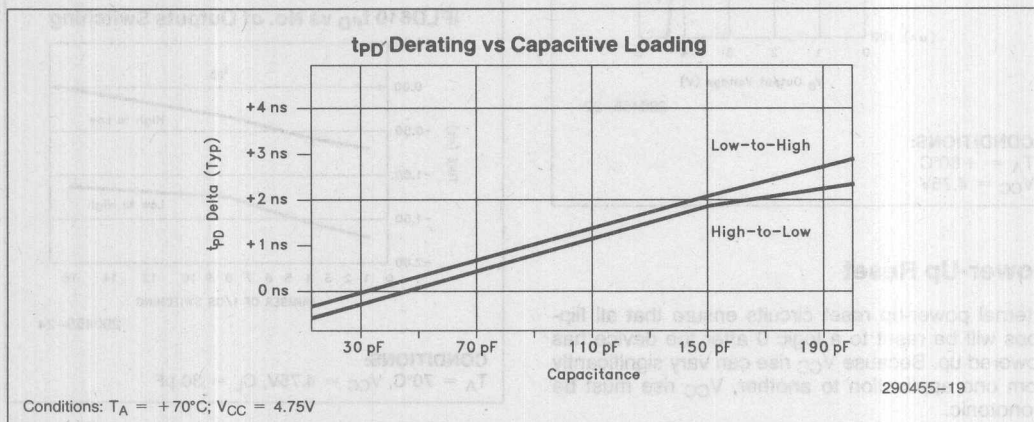
# SYNCHRONOUS REGISTERED MODE



Conditions:  $T_A = 0^\circ\text{C}$ ,  $V_{CC} = 5.25\text{V}$

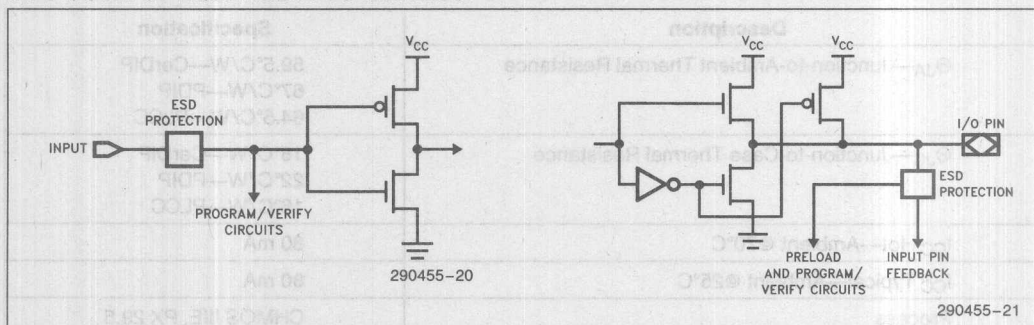
2

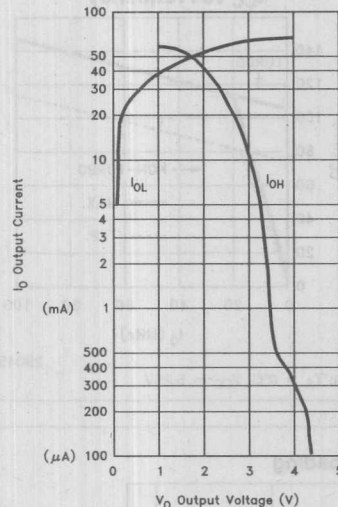
## $t_{PD}$ Derating vs Capacitive Loading



Conditions:  $T_A = +70^\circ\text{C}$ ;  $V_{CC} = 4.75\text{V}$

## INPUT/OUTPUT EQUIVALENT SCHEMATICS



iPLD610 Output Current  
in Relation to Voltage

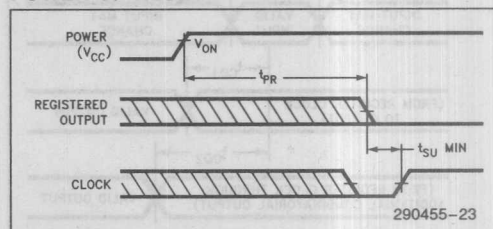
290455-22

**CONDITIONS:** $T_A = +80^\circ\text{C}$  $V_{CC} = 4.75\text{V}$ **Power-Up Reset**

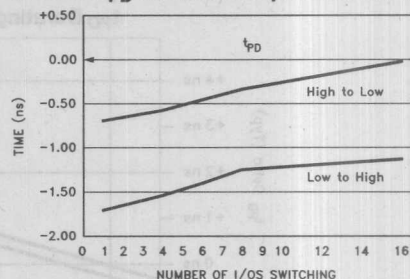
Internal power-up reset circuits ensure that all flip-flops will be reset to a logic 0 after the device has powered up. Because  $V_{CC}$  rise can vary significantly from one application to another,  $V_{CC}$  rise must be monotonic.

**POWER-UP RESET CHARACTERISTICS**

Symbol	Parameter	Value
$t_{PR}$	Power-Up Reset	1000 ns Max.
$V_{ON}$	Turn-On Voltage	4.75V

**POWER-UP RESET**

290455-23

iPLD610  $t_{PD}$  vs No. of Outputs Switching

290455-24

**CONDITIONS:** $T_A = 70^\circ\text{C}$ ,  $V_{CC} = 4.75\text{V}$ ,  $C_L = 30\text{ pF}$ **PACKAGE/TECHNOLOGY SPECIFICATIONS**

Description	Specification
$\Theta_{JA}$ —Junction-to-Ambient Thermal Resistance	59.5°C/W—CerDIP 67°C/W—PDIP 64.5°C/W—PLCC
$\Theta_{JC}$ —Junction-to-Case Thermal Resistance	18°C/W—CerDIP 22°C/W—PDIP 16°C/W—PLCC
$I_{CC}$ Hot—Ambient @70°C	80 mA
$I_{CC}$ Typical—Ambient @25°C	80 mA
Process	CHMOS IIIIE, PX 29.5

This data sheet combines the 85C060 information from 290246-005 and the iPLD610 information from 290246-006. The two parts are architecturally the same.

- Extensive Software and Programming Support via Intel and Third-Party Tools
  - 1-Million CMOS 1E1 EPROM Technology
  - Programmable Low-Power Option for "Standby" Operation: 50  $\mu$ A Typ. in Standby Mode
  - Programmable Security Bit Allows Total Protection of Proprietary Designs
  - 100% Generically Tested Logic Array
  - Available in 40-Pin PDIP and 44-Pin PLCC Packages
  - 85C060 also Available in 40-Pin CERDIP Package
- (See Packaging Notes, Order Number 290246-001, Packages Type II and F)

- 60 to 15 ns, 85.5 MHz W/Feedback, Clock to Output Delay
- $I_{CC} = 150$  mA Max @ 1 MHz
- 24 Macrocells with Programmable I/O Architecture (Register/Combinational)
- Register Configurable as D7/K/R/S Types
- Up to 36 Inputs (12 Dedicated and 24 VIO)
- 8-Input Selectable SOP Invert, Clear, and OE Functions for Each Macrocell
- Programmable Clock System with 2 Synchronous Clocks and Asynchronous Clocking Option on all Macrocells

2

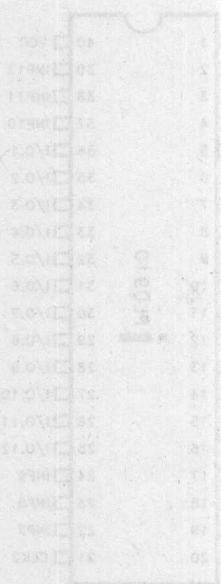
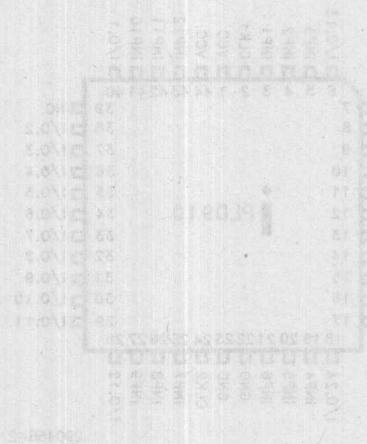


Figure 1 Pinout Diagrams

CH802 is a registered process of Intel Corporation.

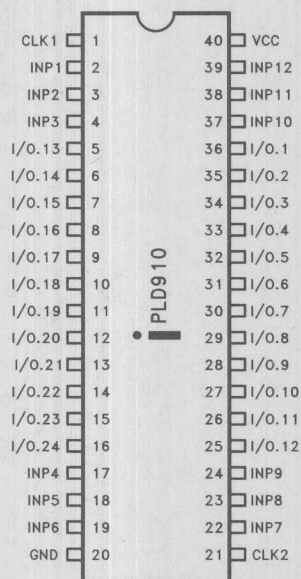


# iPLD910/85C090 FAST 24-MACROCELL CMOS PLD

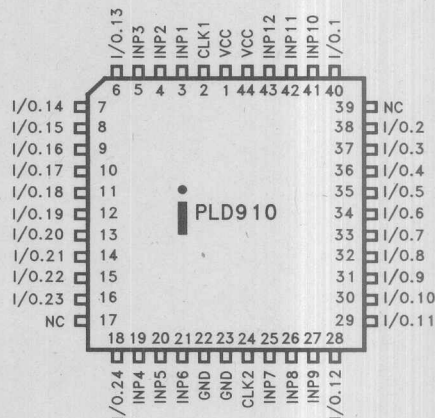
Function, Pin, and JEDEC Compatible with  
EP900, EP910, EP910A, 85C090 and 5C090

- $t_{PD}$  12 ns, 62.5 MHz w/Feedback, Clock to Output 8 ns
- $I_{CC} = 150$  mA Max @ 1 MHz
- 24 Macrocells with Programmable I/O Architecture (Register/Combinatorial). Registers Configurable as D/T/JK/RS Types
- Up to 36 Inputs (12 Dedicated and 24 I/O)
- 8 P-terms, Selectable SOP Invert, Clear and OE P-terms for Each Macrocell
- Programmable Clock System with 2 Synchronous Clocks and Asynchronous Clocking Option on all Macrocells
- Extensive Software and Programming Support via Intel and Third-Party Tools
- 1-Micron CHMOS IIIE\* EPROM Technology
- Programmable Low-Power Option for "Standby" Operation; 60  $\mu$ A Typ. in Standby Mode
- Programmable Security Bit Allows Total Protection of Proprietary Designs
- 100% Generically Tested Logic Array
- Available in 40-Pin PDIP and 44-Pin PLCC Packages
- 85C090 also Available in 40-pin CerDIP Package

(See Packaging Spec., Order Number 240800-001, Package Type N and P)



290456-1



290456-2

Figure 1. Pinout Diagrams

\*CHMOS is a patented process of Intel Corporation.

## 85C090

All information in this document that refers to the iPLD910 is identical to that of the 85C090.

## INTRODUCTION

The iPLD910 is a high-performance, high-integration, general-purpose CMOS PLD. The iPLD910 PLD

(Programmable Logic Device) accommodates logic functions with up to 36 inputs and 24 I/O macrocells. Each I/O macrocell includes 8 product-terms (p-terms) for input, a separate clear p-term, and an output enable/asynchronous clock p-term. With a maximum external frequency of 62.5 MHz, the iPLD910 is well suited to high-performance micro-processor-based systems. The iPLD910 is pin- and function-compatible with the EP900, EP910, EP910A, 85090 and 5C090.

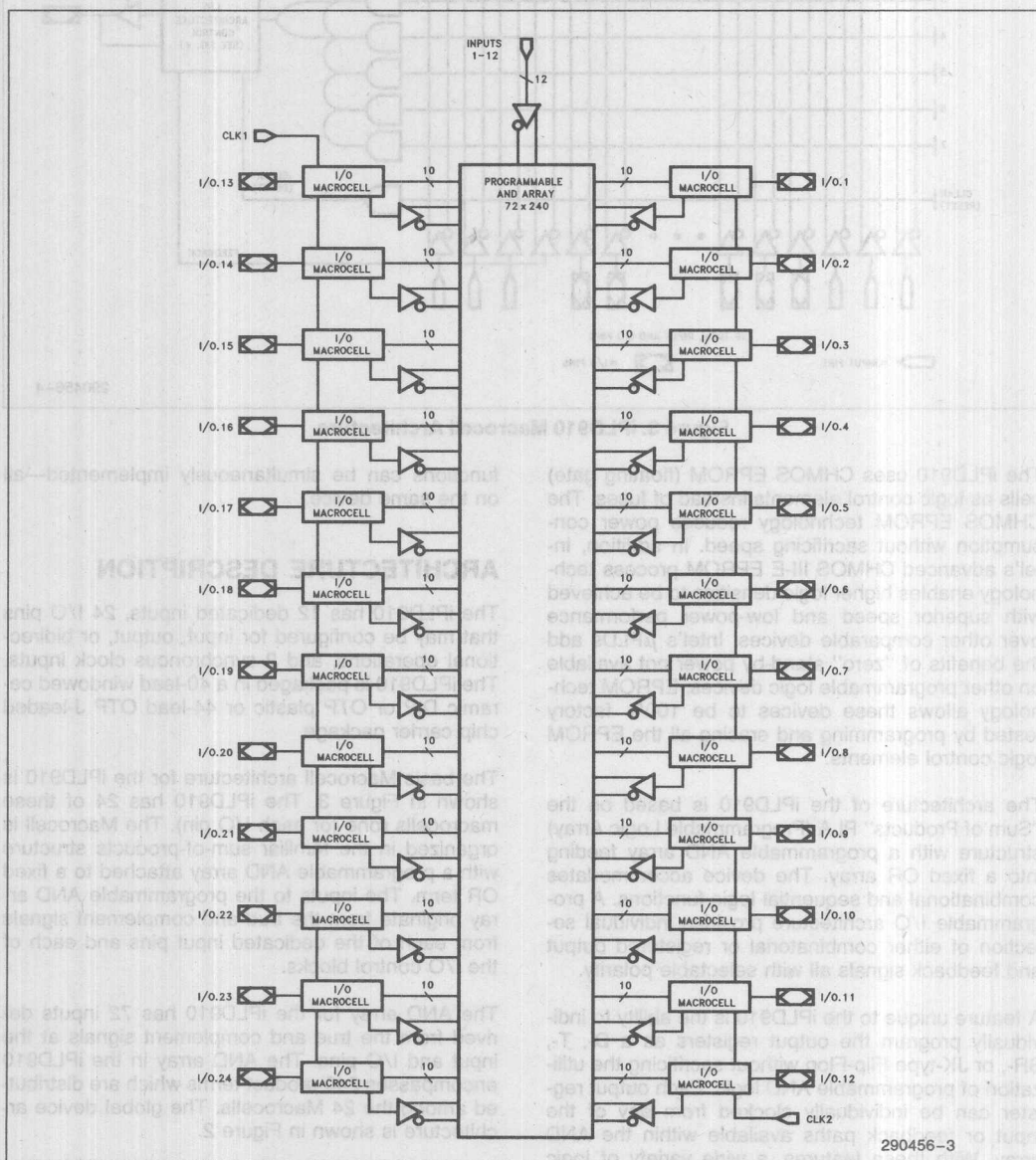


Figure 2. iPLD910 Global Architecture



### Figure 6.11 EBC 15 Master Class Architecture

The AND array for the iPLD910 has 72 inputs derived from the true and complement signals at the input and I/O pins. The AND array in the iPLD910 encompasses 240 product terms which are distributed among the 24 Macrocells. The global device architecture is shown in Figure 2.



Each Macrocell contains ten product terms. Eight of the ten product terms (AND gates) are dedicated for SOP logic implementation. One product term on each Macrocell is used for RESET control to the output register associated with the Macrocell. The final product term is used for OE/Asynchronous Clock implementation.

Within the AND array, there is an EPROM connection at every intersection of an input signal (true and complement) and a product term to a given Macrocell. Before programming an erased device, every EPROM connection is made at every intersection. But during the programming process, these connections are opened so that only the desired connections remain. Therefore, the true or complement of any input signal can be connected to any product term. If both the true and complement connections of any signal are left intact, a logical false results on the output of the AND gate. However, if both the true and complement connections are open, then a logic "don't care" results on the AND gate. Lastly, if all the inputs of a product term are programmed open, then a logical true results on the output of the AND gate.

The iPLD910 has two dedicated clock inputs to provide synchronous clock signals to the internal registers. Each of the clock signals controls half the total registers within the given device. For example, CLK1 provides synchronous clocking to the registers in Macrocells in the left half of the array while CLK2 controls the registers associated with Macrocells in the right half of the array. The advanced I/O architecture allows for any number of the registers to be synchronously clocked (from none to all). Both of the dedicated clock inputs latch the data into a given register when triggered on a positive edge.

## MACROCELL ARCHITECTURE SELECTION

The iPLD910 architecture provides each Macrocell with over 50 different possible I/O register configurations. Each I/O pin can be configured for combinatorial or registered output (true or complement) with feedback. In addition, four different types of output registers can be implemented on I/O pin without any additional logic requirements. The feedback mechanism for each register back into the AND array can be programmed to provide for either registered feedback from the Macrocell or input feedback (treating the pin as an input). Another advantage of the advanced I/O capability of the iPLD910 is the ability to individually clock each internal register from asynchronous clock signals.

## Invert Select EPROM Bit

The Invert Select EPROM bit is used to invert the product term input into the register. This applies to all inputs including double inputs on the JK and SR registers.

## REGISTER SELECTION

The advanced I/O architecture of the iPLD910 allows four different register types along with combinatorial output as illustrated in Figure 4a through e. The register types include a T, D, JK, or SR Flip-Flop and each Macrocell I/O structure may be independently configured. In addition, all registers have an individual asynchronous RESET control from a dedicated product term derived in the AND array. When this dedicated product term is a logical one, the Macrocell register is immediately cleared to a logical zero independent of the register clock. The RESET function occurs automatically on power-up.

## Output Register Configuration

The four different register types shown in Figure 4 are described below.

### D- or T-type Flip-Flops

When either a D- or T-type Flip-Flop is configured as part of the I/O structure, all eight of the product terms into the Macrocell are ORed together and fed into the register input.

### JK or SR Registers

When either a JK or SR register is configured, the eight product terms are shared among two OR gates (one for the J or S input and the other for the K or R input). The allocation for these product terms for each of the register inputs is optimized by the PLDshell Plus software.

## OUTPUT/FEEDBACK

The Output Select Multiplexer allows for either registered, combinatorial or no output.

The Feedback Select Multiplexer EPROM bit enables registered, I/O (using the pin for bidirectional input or just input), or no feedback to the AND array.

The Feedback Select is also important for building equations with more than 8 product terms. The 8-product equations of a Macrocell can be fed back



into the AND array and combined with still more signals to create a much larger product term (of more than 8-inputs). If the feedback product term is not to be used as an output, the associated Macrocell pin should be left floating (no connect) when assembled onto a circuit board.

Any I/O pin may be configured as a dedicated input by selecting no output and pin feedback through the appropriate multiplexers.

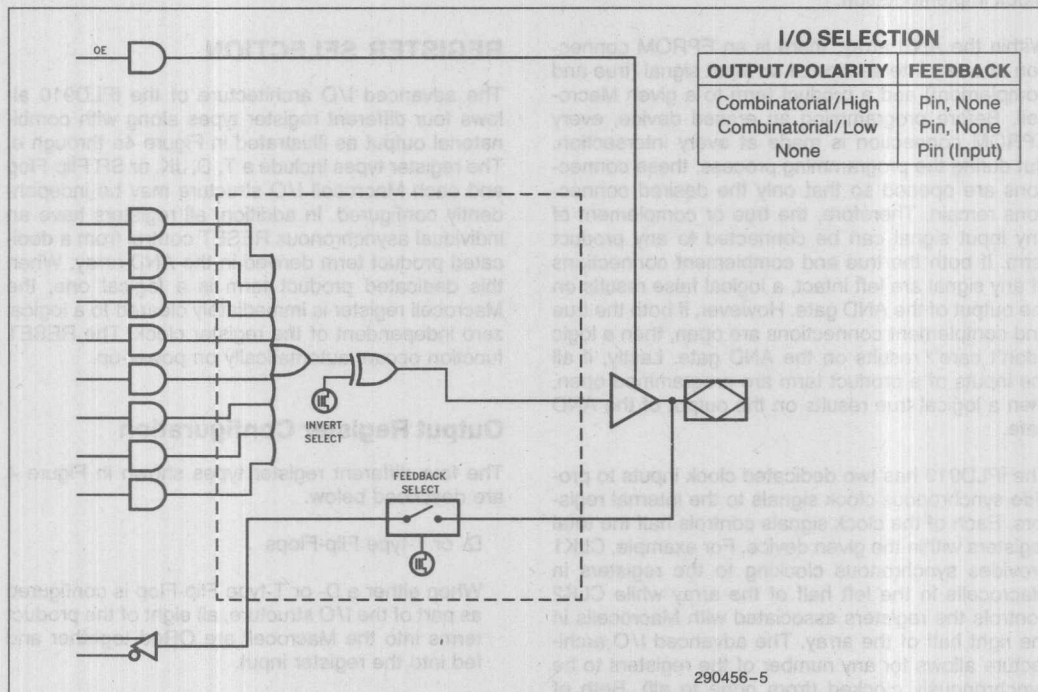


Figure 4a. Combinatorial I/O Configuration

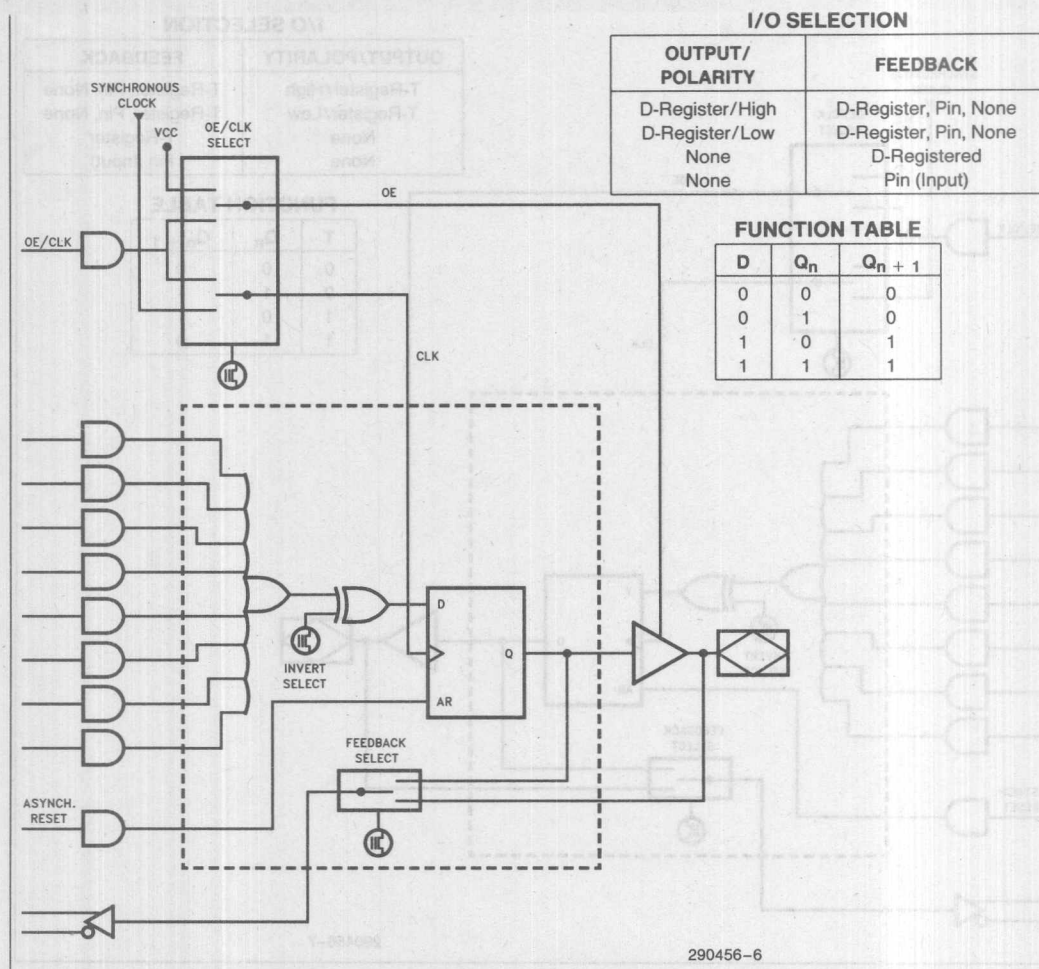


Figure 4b. D-Type Flip-Flop Register Configuration

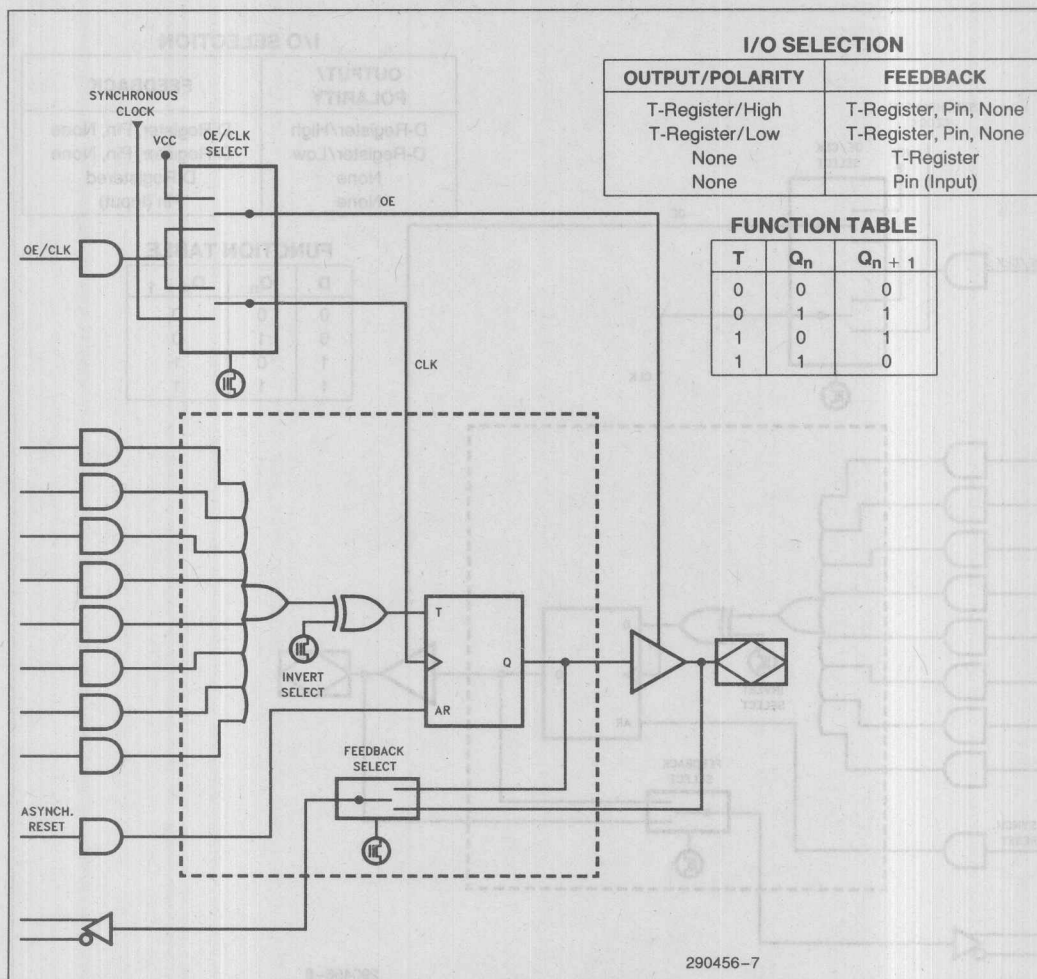


Figure 4c. Toggle Flip-Flop Register Configuration

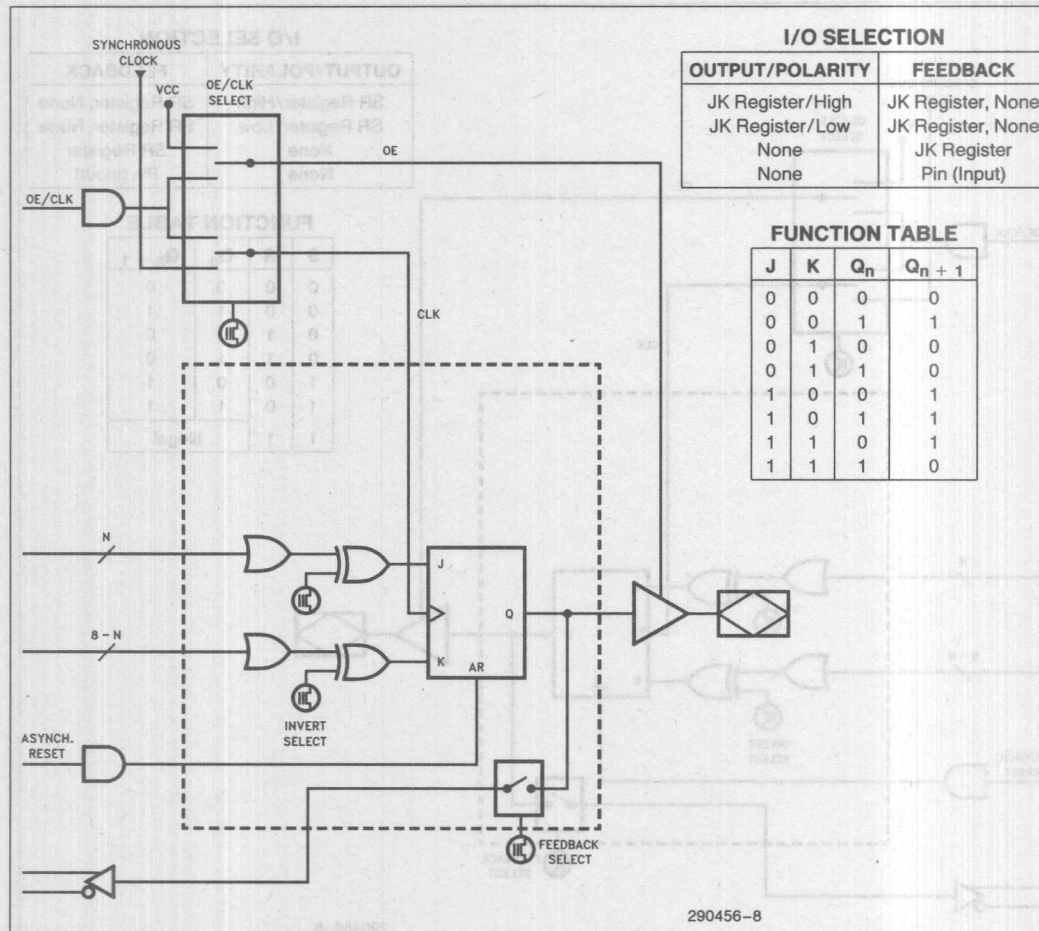


Figure 4d. JK Flip-Flop Register Configuration



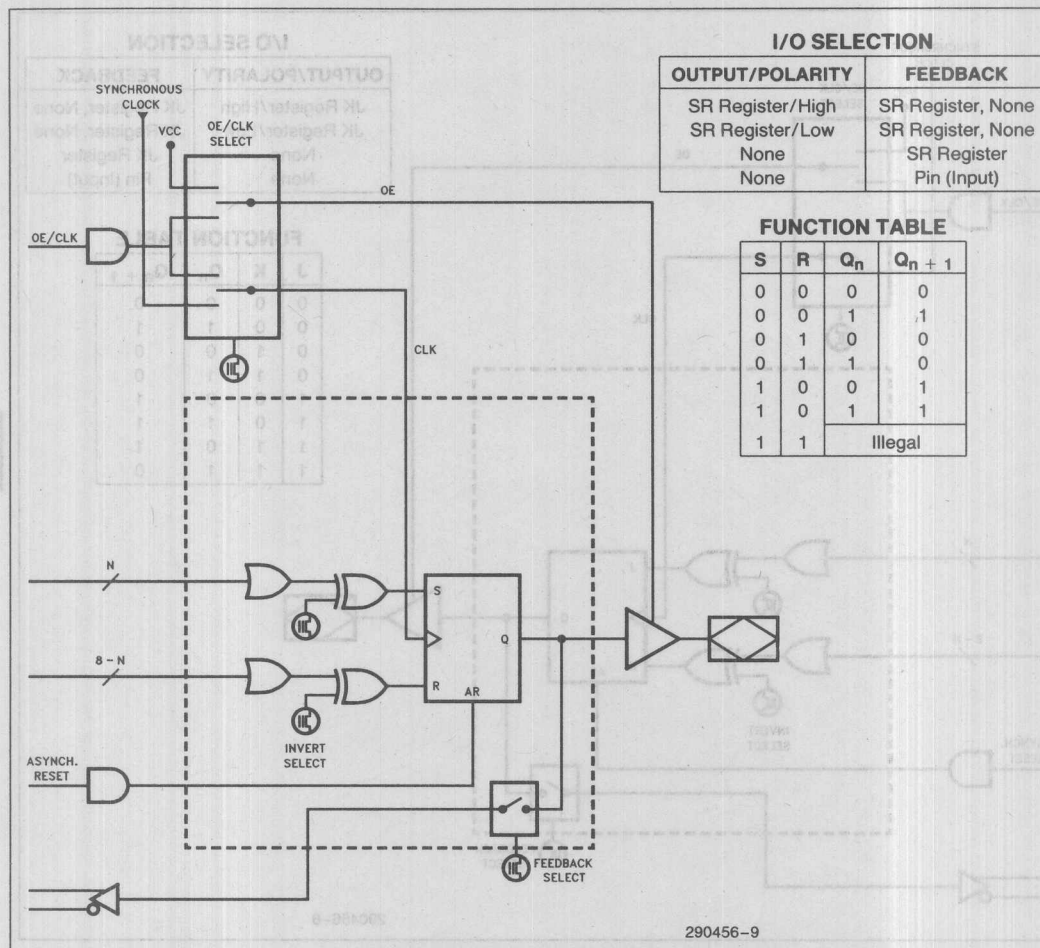


Figure 4e. SR Flip-Flop Register Configuration

## Output Enable (OE)/Clock Selection

Two modes of operation are provided by the OE/CLK Select Multiplexer as a part of each Macrocell. One mode provides for three-state buffering of outputs while in the other mode, the outputs are always enabled. The operation of the OE/CLK Select Multiplexer sets the mode within a given Macrocell. Therefore, the output mode can be selected individually on every output. Figure 5 illustrates the two modes of OE/CLK operation.

### MODE 0: THREE-STATE BUFFERING

In Mode 0, the three-state output buffer is controlled by a single product term originating from the AND array. The output is enabled when the product term is a logical true. Conversely, the output appears as high impedance when the product term is a logical false as shown in Table 1. In Mode 0, the Macrocell Flip-Flop is connected to its associated synchronous clock (either CLK1 or CLK2 depending upon the Macrocell's location within the device). Thus, the Macrocell Flip-Flop may be clocked by its respective synchronous clock but its output will not become valid until the output is enabled.

Table 1. Mode 0 Output Selection

Product Term	Output Buffer
FALSE	Three-State
TRUE	Enabled

### MODE 1: OUTPUT BUFFER ENABLED

In Mode 1, the Output Buffer is always enabled. In addition, the Macrocell Flip-Flop is connected to the AND array. The Macrocell Flip-Flop may now be triggered from an asynchronous clock signal generated by the AND array logic to the OE/CLK multiplexable term. Mode 1 allows the Macrocell Flip-Flops to be individually clocked from any of the available signals in the AND array. Since both true and complement values appear in the AND array, the Flip-Flop may be clocked by positive-or negative-going signals at any input pin. Gated clock structures can be created since the Flip-Flop clock is created by a product term.

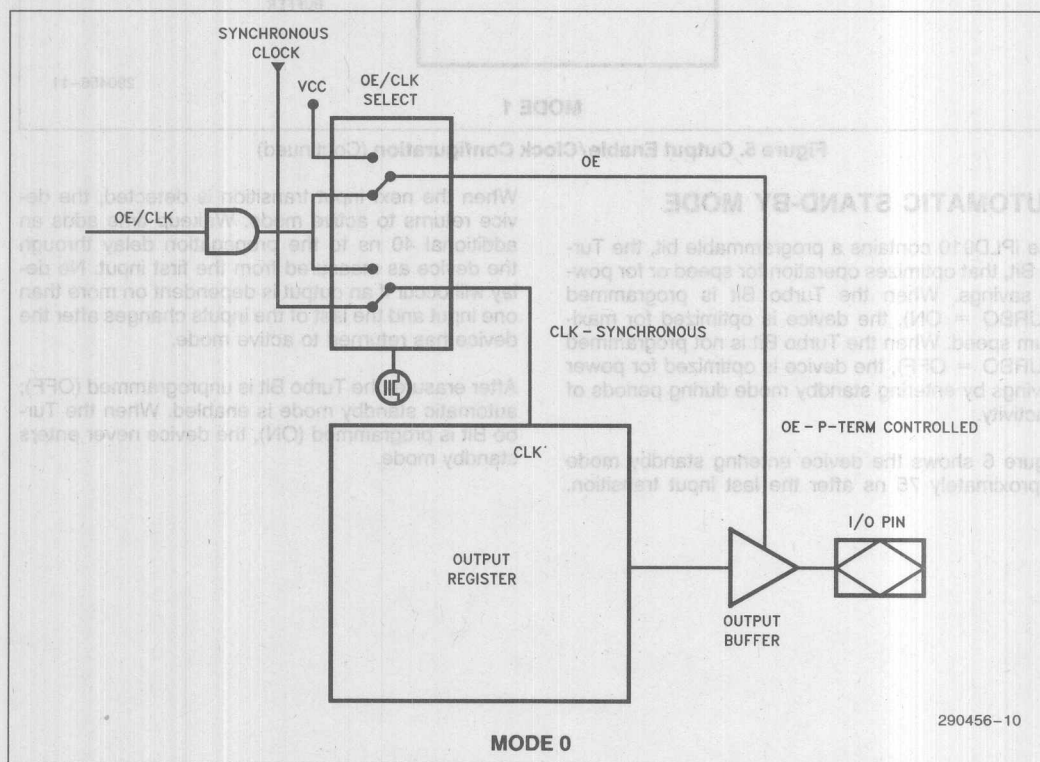


Figure 5. Output Enable/Clock Configuration

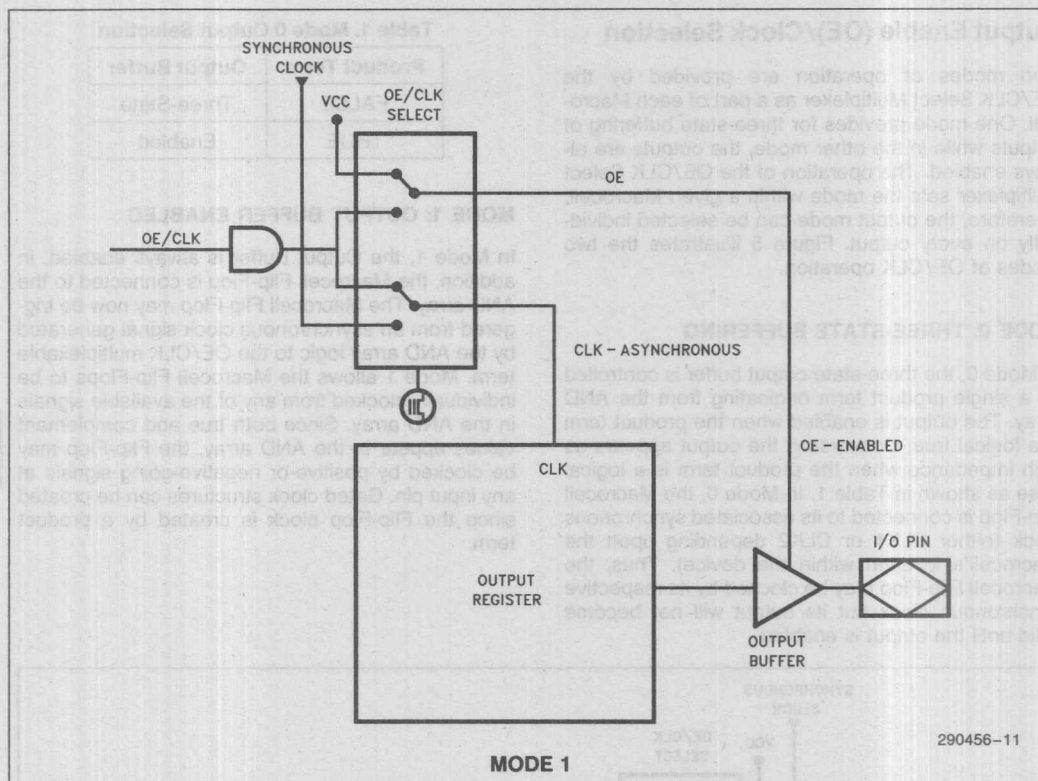


Figure 5. Output Enable/Clock Configuration (Continued)

### AUTOMATIC STAND-BY MODE

The iPLD910 contains a programmable bit, the Turbo Bit, that optimizes operation for speed or for power savings. When the Turbo Bit is programmed (TURBO = ON), the device is optimized for maximum speed. When the Turbo Bit is not programmed (TURBO = OFF), the device is optimized for power savings by entering standby mode during periods of inactivity.

Figure 6 shows the device entering standby mode approximately 75 ns after the last input transition.

When the next input transition is detected, the device returns to active mode. Wakeup time adds an additional 40 ns to the propagation delay through the device as measured from the first input. No delay will occur if an output is dependent on more than one input and the last of the inputs changes after the device has returned to active mode.

After erasure, the Turbo Bit is unprogrammed (OFF); automatic standby mode is enabled. When the Turbo Bit is programmed (ON), the device never enters standby mode.

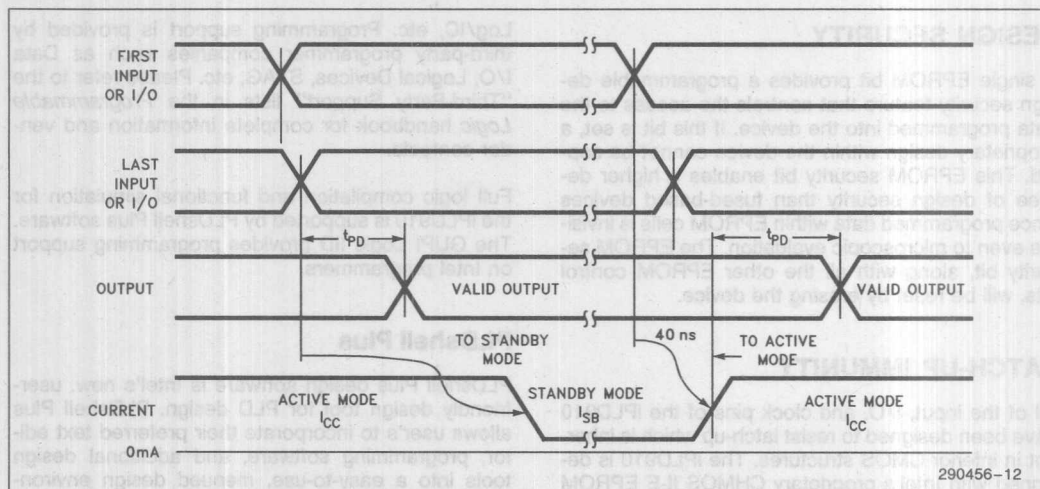


Figure 6. iPLD910 Standby and Active Mode Transitions

## Erased-State Configuration

Prior to programming, the I/O structure is configured for combinatorial active low output with input (pin) feedback.

## PROGRAMMING CHARACTERISTICS

Initially, all the EPROM control bits of the iPLD910 are connected (in the "1" state). Each of the connected control bits are selectively disconnected by programming the EPROM cells into their "0" state. Programming voltage and waveform specifications are available by request from Intel to support programming of the iPLD910.

## Intelligent Programming Algorithm

The iPLD910 supports the Intelligent Programming Algorithm which rapidly programs Intel PLDs while ensuring programming reliability as the incremental program margin of each bit is continually monitored to determine when the bit has been successfully programmed.

## FUNCTIONAL TESTING

Since the logical operation of the iPLD910 is controlled by EPROM elements, the device is completely testable. Each programmable EPROM bit controlling the internal logic is tested using application-independent test program patterns. After testing, the devices are erased before shipment to customers. No post-programming tests of the EPROM array are required.

The testability and reliability of EPROM-based programmable logic devices is an important feature over similar devices based on fuse technology. Fuse-based programmable logic devices require a user to perform post-programming tests to insure proper programming. These tests must be done at the device level because of the cumulative error effect. For example, a board containing ten devices each possessing a 2% device fallout translates into an 18% fallout at the board level (it should be noted that programming fallout of fuse-based programmable logic devices is typically 2% or higher).

## DESIGN RECOMMENDATIONS

For proper operation, it is recommended that all input and output pins be constrained to the voltage range  $GND < (V_{IN} \text{ or } V_{OUT}) < V_{CC}$ . Unused inputs and I/Os should be tied to  $V_{CC}$  or  $GND$  to minimize device power consumption. Reserved pins (as indicated in the logic compiler REPORT file) should be left floating (no connect) so that the pin can attain the appropriate logic level. A power supply decoupling capacitor of at least  $0.2 \mu F$  must be connected directly between  $V_{CC}$  and  $GND$  pins of the device.

As with all CMOS devices, ESD handling procedures should be used with the iPLD910 to prevent damage to the device during programming, assembly and test.



## DESIGN SECURITY

A single EPROM bit provides a programmable design security feature that controls the access to the data programmed into the device. If this bit is set, a proprietary design within the device cannot be copied. This EPROM security bit enables a higher degree of design security than fused-based devices since programmed data within EPROM cells is invisible even to microscopic evaluation. The EPROM security bit, along with all the other EPROM control bits, will be reset by erasing the device.

## LATCH-UP IMMUNITY

All of the input, I/O, and clock pins of the iPLD910 have been designed to resist latch-up which is inherent in inferior CMOS structures. The iPLD910 is designed with Intel's proprietary CHMOS II-E EPROM process. Thus, each of the pins will not experience latch-up with currents up to  $\pm 100$  mA and voltages ranging from  $-1$  V to  $(V_{CC} + 1)$  V. Furthermore, the programming pin is designed to resist latch-up to the 13.5V maximum device limit.

## DEVELOPMENT SOFTWARE

### Third Party Support

The iPLD910 is also supported by third-party logic compilers such as ABEL\*, CUPL\*, PLDesigner\*,

Log/IC, etc. Programming support is provided by third-party programmer companies such as Data I/O, Logical Devices, STAG, etc. Please refer to the "Third-Party Support" lists in the *Programmable Logic* handbook for complete information and vendor contacts.

Full logic compilation and functional simulation for the iPLD910 is supported by PLDshell Plus software. The GUPI Logic IID provides programming support on Intel programmers.

## PLDshell Plus

PLDshell Plus design software is Intel's new, user-friendly design tool for PLD design. PLDshell Plus allows user's to incorporate their preferred text editor, programming software, and additional design tools into a easy-to-use, menued design environment that includes Intel's PLDasm™ logic compiler and simulation software along with disassembly, conversion, and translation utilities. The PLDasm compiler and simulator software accepts industry-standard PDS source files that express designs as Boolean equations, truth tables, or state machines. On-line help, datasheet briefs, technical notes, and error message information, along with waveform viewing/printing capability make the design task as easy as possible. PLDshell Plus software is available from Intel Literature channels or from your local Intel sales representative.

\*ABEL is a trademark of Data I/O Corporation.

CUPL is a trademark of Logical Devices, Inc.

PLDesigner is a trademark of MINC, Inc.

Log/IC™ is a trademark of ISDATA, Corporation.

## ADF PRIMITIVES SUPPORTED

The following ADF primitives are supported by this device:

INP	JOJF
CONF	JONF
COIF	SONF
RONF	SOSF
RORF	TOIF
ROIF	TONF
NORF	TOTF
NOJF	CLKB
NOSF	
NOTF	

## ORDERING INFORMATION

f <sub>CNT1</sub> (MHz)	f <sub>MAX</sub> (MHz)	t <sub>PD</sub> (ns)	Order Code	Package	Operating Range
62.5	100	12	P PLD910-12	PDIP	Commercial
			N PLD910-12	PLCC	Commercial
			D85C090-12	*CerDIP	Commercial
			P85C090-12	PDIP	Commercial
			N85C090-12	PLCC	Commercial
50	83.3	15	P PLD910-15	PDIP	Commercial
			N PLD910-15	PLCC	Commercial
			D85C090-15	*CerDIP	Commercial
			P85C090-15	PDIP	Commercial
			N85C090-15	PLCC	Commercial
33.3	50	25	P PLD910-25	PDIP	Commercial
			N PLD910-25	PLCC	Commercial
			D85C090-25	*CerDIP	Commercial
			P85C090-25	PDIP	Commercial
			N85C090-25	PLCC	Commercial
			TD85C090-25	*CerDIP	Industrial
			TN85C090-25	PLCC	Industrial

Commercial: 0°C to +70°C

Industrial: -40°C to +85°C

\*Windowed package allows UV erase.

**ABSOLUTE MAXIMUM RATINGS\***

Symbol	Parameter	Min	Max	Units
V <sub>CC</sub>	Supply Voltage(1)	-2.0	7.0	V
V <sub>PP</sub>	Programming Supply Voltage(1)	-2.0	13.5	V
V <sub>I</sub>	DC Input Voltage(1)(2)	-0.5	V <sub>CC</sub> + 0.5	V
t <sub>stg</sub>	Storage Temperature	-65	+150	°C
t <sub>amb</sub>	Ambient Temperature(3)	-10	+85	°C

**NOTES:**

1. Voltages with respect to ground.
2. Minimum DC input is -0.5V. During transitions, the inputs may undershoot to -2.0V or overshoot to +7.0V for periods less than 20 ns under no load conditions.
3. Under bias. Extended temperature versions are also available.

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

*\*WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

**RECOMMENDED OPERATING CONDITIONS**

Symbol	Parameter	Min	Max	Unit
V <sub>CC</sub>	Supply Voltage	4.75	5.25	V
V <sub>IN</sub>	Input Voltage	0	V <sub>CC</sub>	V
V <sub>O</sub>	Output Voltage	0	V <sub>CC</sub>	V
T <sub>A</sub>	Operating Temperature	0	+70	°C
t <sub>R</sub>	Input Rise Time		500	ns
t <sub>F</sub>	Input Fall Time		500	ns

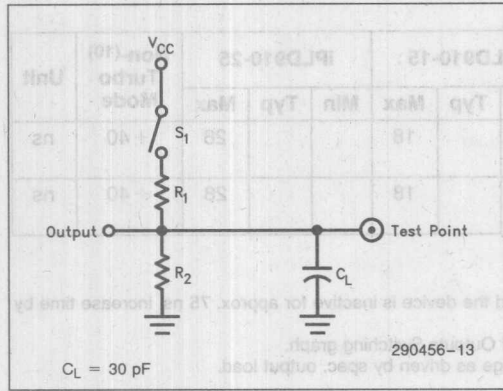
**D.C. CHARACTERISTICS** T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5.0V ±5%

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
V <sub>IH</sub> (4)	High Level Input Voltage	2.0		V <sub>CC</sub> + 0.3	V	
V <sub>IL</sub> (4)	Low Level Input Voltage	-0.3		0.8	V	
V <sub>OH</sub>	High Level Output Voltage	2.4			V	I <sub>O</sub> = -4.0 mA D.C., V <sub>CC</sub> = min.
V <sub>OL</sub> (5)	Low Level Output Voltage			0.45	V	I <sub>O</sub> = 12.0 mA D.C., V <sub>CC</sub> = min.
I <sub>I</sub>	Input Leakage Current	-10		+10	μA	V <sub>CC</sub> = max., GND < V <sub>IN</sub> < V <sub>CC</sub>
I <sub>OZ</sub>	Output Leakage Current	-10		+10	μA	V <sub>CC</sub> = max., GND < V <sub>OUT</sub> < V <sub>CC</sub>
I <sub>SC</sub> (6)	Output Short Circuit Current	-30		-120	mA	V <sub>CC</sub> = max., V <sub>OUT</sub> = 0.5V
I <sub>SB</sub> (7)	Standby Current		60	150	μA	V <sub>CC</sub> = max., V <sub>IN</sub> = V <sub>CC</sub> or GND, Standby Mode
I <sub>CC</sub>	Power Supply Current (See I <sub>CC</sub> vs. Freq. Graph)		4	12	mA	V <sub>CC</sub> = max., V <sub>IN</sub> = V <sub>CC</sub> or GND, No Load, f <sub>IN</sub> = 1 MHz, Device Prog. as Two 12-Bit Counters, Turbo = Off
			120	150	mA	Turbo = On, f <sub>IN</sub> = 1 MHz
I <sub>CCI</sub>	Power Supply Current Industrial Temperature			180	mA	Turbo = On, f <sub>IN</sub> = 1 MHz

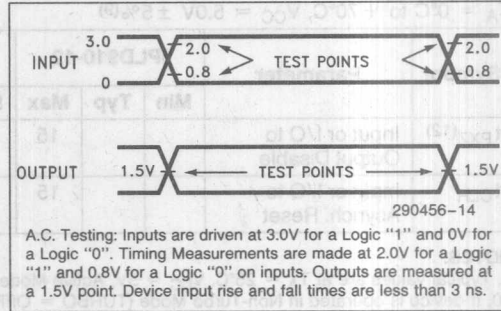
**NOTES:**

4. Absolute values with respect to device GND; all over and undershoots due to system or tester noise are included.
5. Maximum DC I<sub>OL</sub> for the device is 96 mA for CLK1 group I/O.1-I/O.12 and 96 mA for CLK2 group I/O.13-I/O.24.
6. Not more than 1 output should be tested at a time. Duration of that test should not exceed 1 second.
7. In Non-Turbo Mode (TURBO = OFF), device enters standby mode approximately 75 ns after the last input transition.

### A.C. TESTING LOAD CIRCUIT



### A.C. TESTING INPUT, OUTPUT WAVEFORM



2

### SWITCHING TEST CIRCUIT

Specification	$S_1$	$C_L$	Commercial		Measured Output Value
			$R_1$	$R_2$	
$t_{PD}$	Closed	30 pF	200 $\Omega$	330 $\Omega$	1.5V
$t_{PZX}$	Z $\rightarrow$ H: Open Z $\rightarrow$ L: Closed				1.5V
$t_{PXZ}$	H $\rightarrow$ Z: Open L $\rightarrow$ Z: Closed	5 pF			H $\rightarrow$ Z: $V_{OH} - 0.5V$ L $\rightarrow$ Z: $V_{OL} + 0.5V$

**CAPACITANCE**  $T_A = 0^\circ\text{C to } +70^\circ\text{C}; V_{CC} = 5.0V \pm 5\%$ (8)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$C_{IN}$	Input Capacitance	$V_{IN} = 0V, f = 1.0 \text{ MHz}$		5	8	pF
$C_{IO}$	I/O Capacitance	$V_{OUT} = 0V, f = 1.0 \text{ MHz}$		6	8	pF
$C_{CLK}$	CLK Capacitance	$V_{OUT} = 0V, f = 1.0 \text{ MHz}$		8	10	pF
$C_{VPP}$	$V_{PP}$ Pin Capacitance	$V_{PP}$ on CLK2, $f = 1.0 \text{ MHz}$		10	12	pF

**NOTES:**

8. These values are evaluated at initial characterization and whenever design modifications occur that may affect capacitance.

### COMBINATORIAL MODE A.C. CHARACTERISTICS

$T_A = 0^\circ\text{C to } +70^\circ\text{C}, V_{CC} = 5.0V \pm 5\%$ (9)

Symbol	Parameter	iPLD910-12			iPLD910-15			iPLD910-25			Non-(10) Turbo Mode	Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
$t_{PD}^{(11)}$	Input or I/O to Output Valid w/ 8 Outputs Switching			12			15			25	+40	ns
$t_{PZX}^{(12)}$	Input or I/O to Output Enable			15			18			28	+40	ns



**COMBINATORIAL MODE A.C. CHARACTERISTICS** (Continued)T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5.0V ± 5%<sup>(9)</sup>

Symbol	Parameter	iPLD910-12			iPLD910-15			iPLD910-25			Non-(10) Turbo Mode	Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
t <sub>PXZ</sub> <sup>(12)</sup>	Input or I/O to Output Disable			15			18			28	+ 40	ns
t <sub>CLR</sub>	Input or I/O to Asynch. Reset			15			18			28	+ 40	ns

**NOTES:**9. Typical values are at T<sub>A</sub> = 25°C, V<sub>CC</sub> = 5V, Active Mode.

10. If device is operated in Non-Turbo Mode (TURBO = OFF) and the device is inactive for approx. 75 ns, increase time by amount shown.

11. Measured with eight outputs switching. See t<sub>PD</sub> vs. Number of Outputs Switching graph.12. t<sub>PZX</sub> and t<sub>PXZ</sub> are measured at ±0.5V from steady state voltage as driven by spec. output load.**REGISTER MODE—SYNCHRONOUS CLOCK A.C. CHARACTERISTICS**T<sub>A</sub> = 0°C to 70°C, V<sub>CC</sub> = 5.0V ± 5%<sup>(9)</sup>

Symbol	Parameter	iPLD910-12			iPLD910-15			iPLD910-25			Non-(10) Turbo Mode	Units
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
f <sub>CNT1</sub> <sup>(13)</sup>	Max. Counter Frequency 1/(t <sub>SU</sub> + t <sub>CO</sub> )—Ext. Feedback	62.5	75		50	66		33	40			MHz
f <sub>CNT2</sub> <sup>(13)</sup>	Max. Counter Frequency 1/(t <sub>CNT</sub> )—Internal Feedback	76.9	85		66.6	75		40	50			MHz
f <sub>MAX</sub>	Max. Frequency (Pipelined) 1/(t <sub>CW</sub> )—No Feedback	100	110		83.3	100		50	66			MHz
t <sub>SU</sub>	Input or I/O Setup Time to CLK	8			11			16			+ 40	ns
t <sub>H</sub>	Input or I/O Hold Time from CLK	0			0			0				ns
t <sub>CO1</sub> <sup>(13)</sup>	CLK High to Output Valid			8			9			14		ns
t <sub>CO2</sub>	CLK High to Output Valid Fed Through Comb. Macrocell			17			20			30	+ 40	ns
t <sub>CNT</sub> <sup>(13)</sup>	Macrocell Output Feedback to Macrocell Input—Internal Path			13			15			25	+ 40	ns
t <sub>CL</sub>	CLK Low Time	4			5			8				ns
t <sub>CH</sub>	CLK High Time	4			5			8				ns
t <sub>CP</sub>	CLK Period	10			12			20				ns

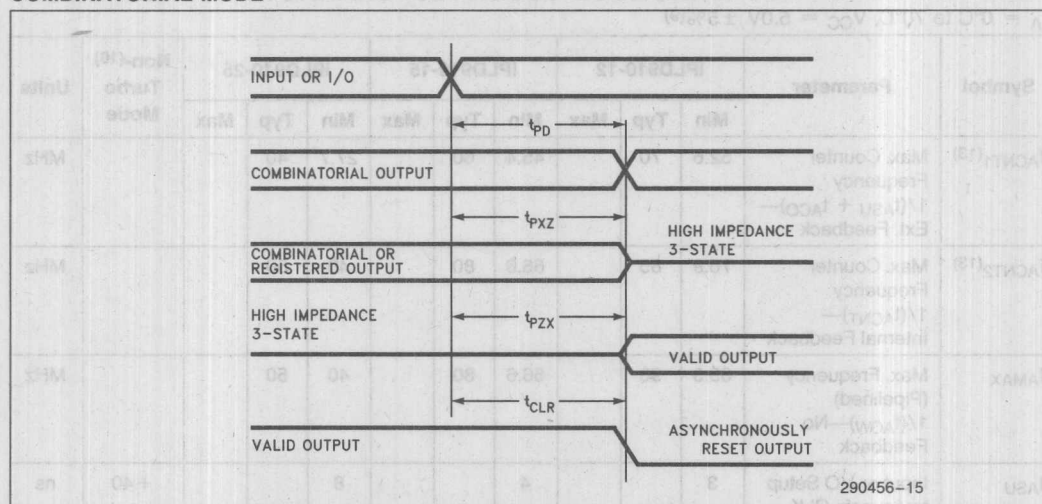
**NOTE:**

13. Measured with device configured as 24-bit counter.

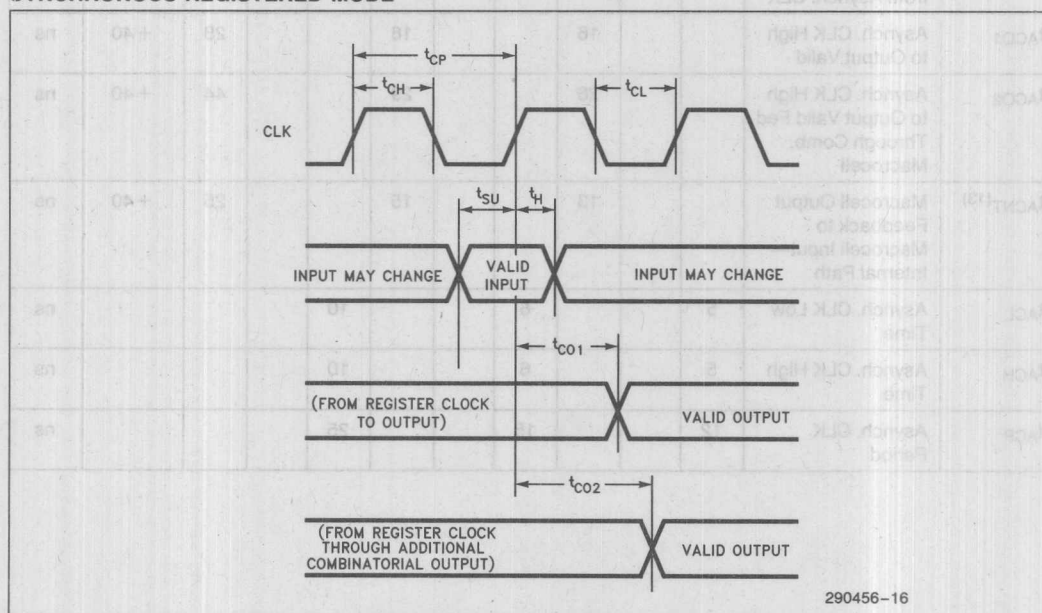
**REGISTER MODE—ASYNCHRONOUS CLOCK A.C. CHARACTERISTICS** $T_A = 0^\circ\text{C to } 70^\circ\text{C}$ ,  $V_{CC} = 5.0\text{V} \pm 5\%$ (9)

Symbol	Parameter	iPLD910-12			iPLD910-15			iPLD910-25			Non-(10) Turbo Mode	Units
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
$f_{ACNT1}^{(13)}$	Max. Counter Frequency $1/(t_{ASU} + t_{ACO})$ — Ext. Feedback	52.6	70		45.4	60		27.7	40			MHz
$f_{ACNT2}^{(13)}$	Max. Counter Frequency $1/(t_{ACNT})$ — Internal Feedback	76.9	85		66.6	80		40	50			MHz
$f_{AMAX}$	Max. Frequency (Pipelined) $1/(t_{ACW})$ —No Feedback	83.3	90		66.6	80		40	50			MHz
$t_{ASU}$	Input or I/O Setup to Asynch. CLK	3			4			8			+ 40	ns
$t_{AH}$	Input or I/O Hold from Asynch. CLK	6			7			8				ns
$t_{ACO1}$	Asynch. CLK High to Output Valid			16			18			28	+ 40	ns
$t_{ACO2}$	Asynch. CLK High to Output Valid Fed Through Comb. Macrocell			26			29			44	+ 40	ns
$t_{ACNT}^{(13)}$	Macrocell Output Feedback to Macrocell Input— Internal Path			13			15			25	+ 40	ns
$t_{ACL}$	Asynch. CLK Low Time	5			6			10				ns
$t_{ACH}$	Asynch. CLK High Time	5			6			10				ns
$t_{ACP}$	Asynch. CLK Period	12			15			25				ns

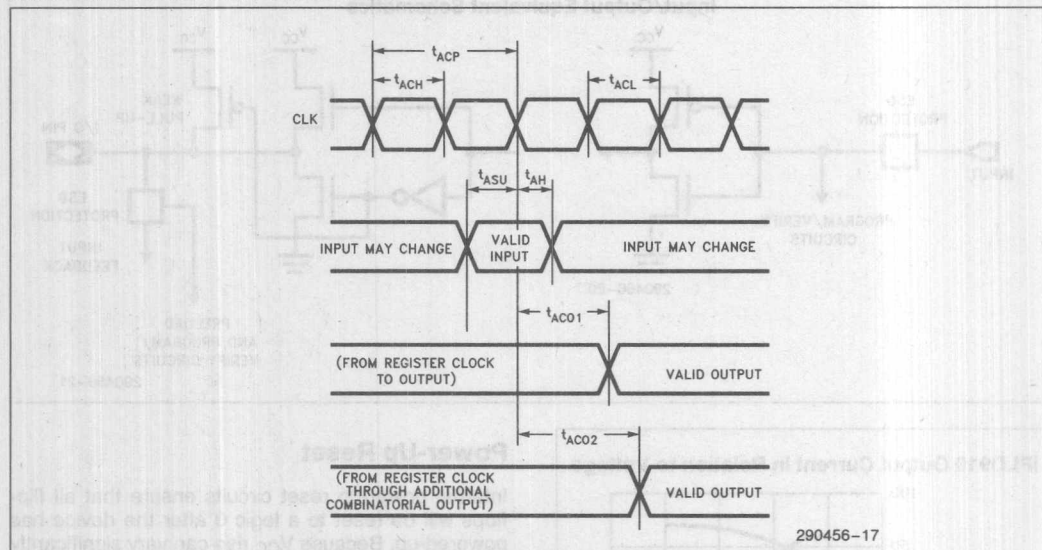
# COMBINATORIAL MODE



# SYNCHRONOUS REGISTERED MODE



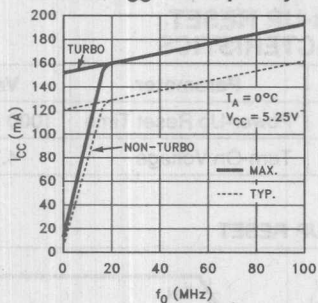
# ASYNCHRONOUS REGISTERED MODE



290456-17

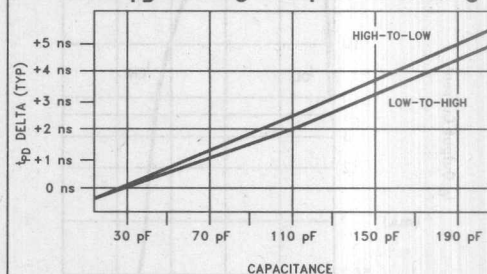
2

iPLD910  $I_{CC}$  vs Frequency



290456-18

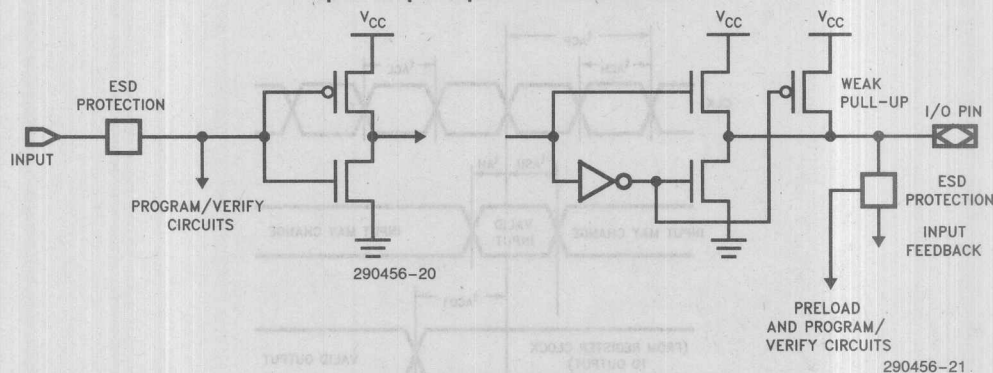
iPLD910  $t_{PD}$  Derating vs Capacitive Loading



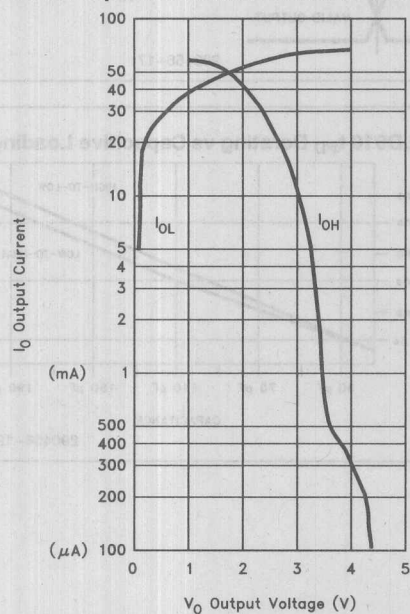
290456-19



### Input/Output Equivalent Schematics



### IPLD910 Output Current in Relation to Voltage



290456-22

#### CONDITIONS:

$T_A = +70^\circ C$

$V_{CC} = 4.75V$

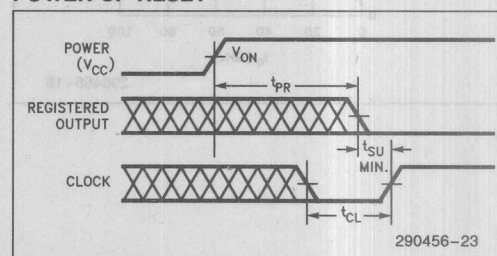
### Power-Up Reset

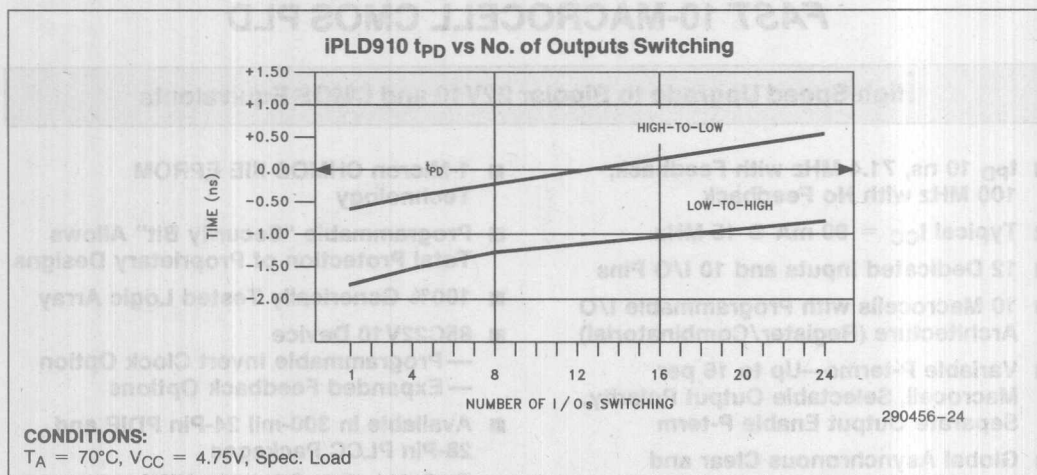
Internal power-up reset circuits ensure that all flip-flops will be reset to a logic 0 after the device has powered-up. Because  $V_{CC}$  rise can vary significantly from one application to another,  $V_{CC}$  rise must be monotonic.

### POWER-UP RESET CHARACTERISTICS

Symbol	Parameter	Value
$t_{PR}$	Power-Up Reset Time	1000 ns Max.
$V_{ON}$	Turn-On Voltage	4.75V

### POWER-UP RESET





2

**PACKAGE/TECHNOLOGY SPECIFICATIONS**

Description	Specification
$\theta_{JA}$ —Junction-to-Ambient Thermal Resistance	44.5°C/W-CerDIP 51°C/W-PDIP 55°C/W-PLCC
$\theta_{JC}$ —Junction-to-Case Thermal Resistance	17°C/W-CerDIP 29°C/W-PDIP 16°C/W-PLCC
$I_{CC}$ Hot—Ambient @70°C	125 mA
$I_{CC}$ Typical—Ambient @25°C	125 mA
Process	CHMOS IIIIE, PX 29.5



## iPLD22V10 FAST 10-MACROCELL CMOS PLD

High-Speed Upgrade to Bipolar 22V10 and CMOS Equivalents

- $t_{PD}$  10 ns, 71.4 MHz with Feedback, 100 MHz with No Feedback
- Typical  $I_{CC} = 90$  mA @ 15 MHz
- 12 Dedicated Inputs and 10 I/O Pins
- 10 Macrocells with Programmable I/O Architecture (Register/Combinatorial)
- Variable P-terms—Up to 16 per Macrocell, Selectable Output Polarity, Separate Output Enable P-term
- Global Asynchronous Clear and Synchronous Preset P-terms
- 1-Micron CHMOS III E EPROM Technology
- Programmable "Security Bit" Allows Total Protection of Proprietary Designs
- 100% Generically Tested Logic Array
- 85C22V10 Device
  - Programmable Invert Clock Option
  - Expanded Feedback Options
- Available in 300-mil 24-Pin PDIP and 28-Pin PLCC Packages

(See Packaging Spec., Order Number 240800, Package Type N and P)

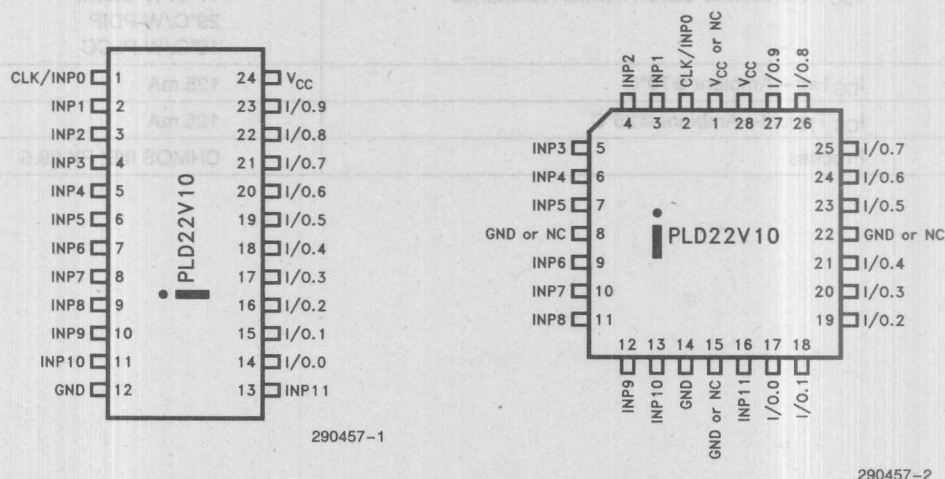


Figure 1. Pinout Diagrams

## INTRODUCTION

The iPLD22V10 is a high-performance, high-integration, general-purpose CMOS PLD. The iPLD22V10 accommodates logic functions with up to 22 inputs and 10 I/O macrocells. I/O macrocells include an average of 12 p-terms for input, with a separate p-term for output enable. Figure 2 shows the global architecture of the device.

## JEDEC AND PIN COMPATIBILITY

The iPLD22V10 is 100% JEDEC-, pin- and function-compatible with the industry-standard 22V10 PLD. JEDEC files developed for 22V10 devices can be used to program the iPLD22V10. When the N PLD22V10 (28-pin PLCC) is used to replace a conventional 22V10 in an existing design socket, pins 8, 15, 22 and 1 are left as No Connects (NC). New designs can take advantage of the additional device  $V_{CC}$  and grounds these pins offer.

## PROGRAMMABLE MACROCELLS

In addition to the 12 dedicated input pins, the iPLD22V10 contains 10 programmable macrocells. Each of the macrocells can be programmed to function as an input or as a combinatorial or registered output. Programmable output polarity and programmable feedback options allow the iPLD22V10 to be tailored to the precise needs of the target application. Figure 3 shows the architecture of each macrocell.

## Output Polarity

The output polarity for each iPLD22V10 macrocell is programmable. Each combinatorial or registered output can be active-high or active-low.

## ORDERING INFORMATION

$f_{CNT1}$ (MHz)	$f_{MAX}$ (MHz)	$t_{PD}$ (ns)	Order Code	Package	Operating Range
71.4	100	10	P PLD22V10-10	PDIP	Commercial
			N PLD22V10-10	PLCC	Commercial
58.8	83.3	15	P PLD22V10-15	PDIP	Commercial
			N PLD22V10-15	PLCC	Commercial
35.7	40	25	P PLD22V10-25	PDIP	Commercial
			N PLD22V10-25	PLCC	Commercial

## Feedback Options

iPLD22V10 macrocells programmed as combinatorial outputs support pin feedback to the logic array (i.e., feedback from the I/O pin). iPLD22V10 macrocells programmed as registers allow internal register feedback to the logic array.

## 85C22V10 SUPERSET DEVICE

Intel offers a 22V10 architecture superset device, the 85C22V10. These architecture superset features include invertible clock and expanded feedback options. Designs may be developed for the 85C22V10 using Intel's PLDshell plus or popular third party tools such as ABEL. For complete details on the 85C22V10, refer to the Datasheet (Lit. #290416).

## Expanded Feedback Options (85C22V10)

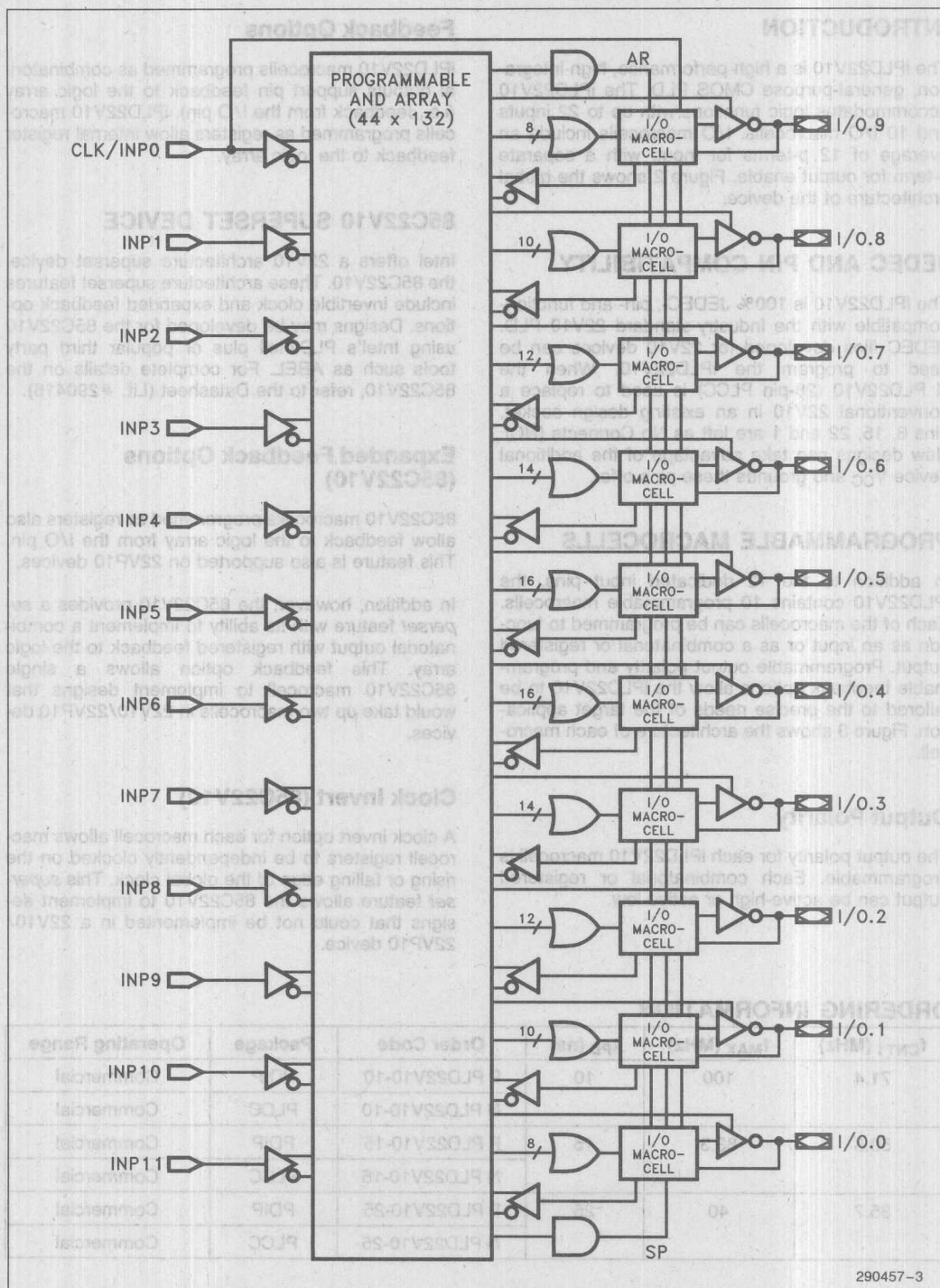
85C22V10 macrocells programmed as registers also allow feedback to the logic array from the I/O pin. This feature is also supported on 22VP10 devices.

In addition, however, the 85C22V10 provides a *superset* feature with its ability to implement a combinatorial output with registered feedback to the logic array. This feedback option allows a single 85C22V10 macrocell to implement designs that would take up two macrocells in 22V10/22VP10 devices.

## Clock Invert (85C22V10)

A clock invert option for each macrocell allows macrocell registers to be independently clocked on the rising or falling edge of the global clock. This *superset* feature allows the 85C22V10 to implement designs that could not be implemented in a 22V10/22VP10 device.





290457-3

Figure 2. iPLD22V10 Global Architecture

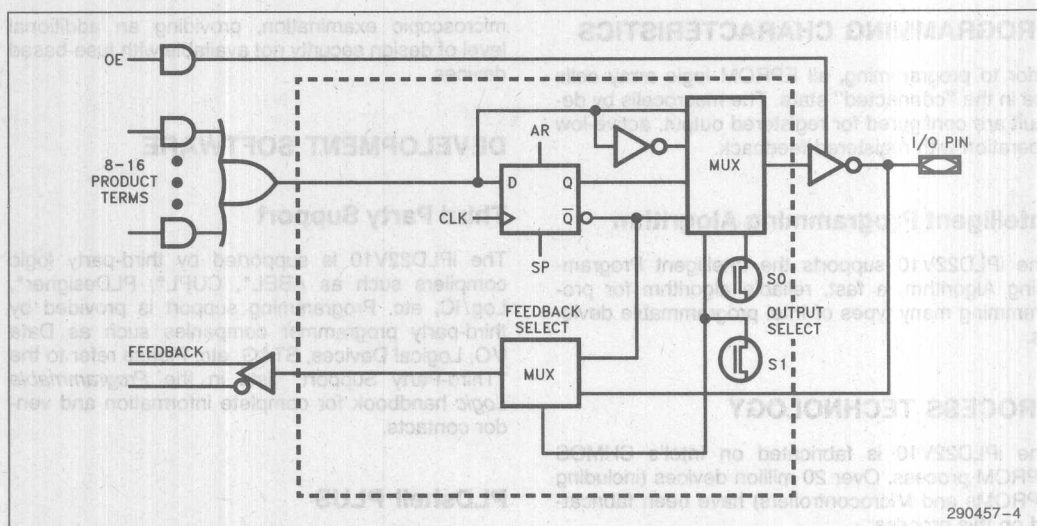


Figure 3. iPLD22V10 Macrocell Architecture

Table 1 lists the macrocell configurations:

Table 1. iPLD22V10 Macrocell Configurations

S1	S0	Output/Polarity	Feedback
0	0	Registered/Active Low	Registered
0	1	Registered/Active High	Registered
1	0	Combinatorial/Active Low	Pin
1	1	Combinatorial/Active High	Pin

### Register Preset/Reset

iPLD22V10 macrocell registers can be preset or reset using global preset and reset p-terms. Register preset is synchronous and must meet the specified setup time to the clock signal. Register reset is asynchronous and has no setup requirement to the clock. Preset and reset set or reset the register. Output polarity is selected separately.

### Programmable Output Enable

Each macrocell contains an output buffer that can place the respective output in a high-impedance state (three-state). The output buffer is controlled by a single p-term per macrocell in the logic array and is asynchronous.

### POWER-ON CHARACTERISTICS

iPLD22V10 inputs and outputs begin responding 1  $\mu$ s (max.) after  $V_{CC}$  power-up ( $V_{CC} = 4.75V$ ) or after a power-loss/power-up sequence. All macrocells programmed as registers are set to a logic low.

## PROGRAMMING CHARACTERISTICS

Prior to programming, all EPROM logic array cells are in the "connected" state. The macrocells by default are configured for registered output, active-low operation with registered feedback.

### Intelligent Programming Algorithm

The iPLD22V10 supports the Intelligent Programming Algorithm, a fast, reliable algorithm for programming many types of Intel programmable devices.

## PROCESS TECHNOLOGY

The iPLD22V10 is fabricated on Intel's CHMOS EPROM process. Over 20 million devices (including EPROMs and Microcontrollers) have been fabricated on this process.

## TESTABILITY

The iPLD22V10 is completely tested at the factory. Unlike fuse-based PLDs, which have one-time programmable fuse links that limit testing to small-scale sampling, each EPROM cell in the iPLD22V10 is tested and erased prior to shipment.

## SECURITY

A single programmable bit, called the security bit or verify protect bit, controls access to the data programmed into the device. Once this security bit is set, the design cannot be copied.

Since data in the device is stored in EPROM cells, the contents of the device cannot be read even with

microscopic examination, providing an additional level of design security not available with fuse-based devices.

## DEVELOPMENT SOFTWARE

### Third Party Support

The iPLD22V10 is supported by third-party logic compilers such as ABEL\*, CUPL\*, PLDesigner\*, Log/IC, etc. Programming support is provided by third-party programmer companies such as Data I/O, Logical Devices, STAG, etc. Please refer to the "Third-Party Support" lists in the *Programmable Logic* handbook for complete information and vendor contacts.

### PLDshell PLUS

Full logic compilation and functional simulation for the iPLD22V10 is supported by PLDshell Plus software.

PLDshell Plus design software is Intel's new, user-friendly design tool for PLD design. PLDshell Plus allows users to incorporate their preferred text editor, programming software, and additional design tools into a easy-to-use, menued design environment that includes Intel's PLDasm™ logic compiler and simulation software along with disassembly, conversion, and translation utilities. The PLDasm compiler and simulator software accepts industry-standard PDS source files that express designs as Boolean equations, truth tables, or state machines. On-line help, datasheet briefs, technical notes, and error message information, along with waveform viewing/printing capability make the design task as easy as possible. PLDshell Plus software is available from Intel Literature channels or from your local Intel sales representative, order #468810.

\*ABEL is a trademark of Data I/O Corp. CUPL is a trademark of Logical Devices, Inc. PLDesigner is a trademark of MINC, Inc. Log/IC is a trademark of ISDATA, Corporation.

**ABSOLUTE MAXIMUM RATINGS\***

Supply Voltage ( $V_{CC}$ )<sup>(1)</sup> ..... -2.0V to +7.0V  
 Programming Supply  
 Voltage ( $V_{PP}$ )<sup>(1)</sup> ..... -2.0V to +13.5V  
 D.C. Input Voltage ( $V_I$ )<sup>(1,2)</sup> ... -0.5V to  $V_{CC} + 0.5V$   
 Storage Temperature ( $T_{stg}$ ) ..... -65°C to +150°C  
 Ambient Temperature ( $T_A$ )<sup>(3)</sup> ..... -10°C to +85°C

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

*\*WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

**RECOMMENDED OPERATING CONDITIONS**

Symbol	Parameter	Min	Max	Units
$V_{CC}$	Supply Voltage	4.75	5.25	V
$V_{IN}$	Input Voltage	0	$V_{CC}$	V
$V_O$	Output Voltage	0	$V_{CC}$	V
$T_A$	Operating Temperature	0	+70	°C
$t_R$	Input Rise Time		500	ns
$t_F$	Input Fall Time		500	ns

2

**D.C. CHARACTERISTICS** ( $T_A = 0^\circ\text{C}$  to  $+70^\circ\text{C}$ ,  $V_{CC} = 5.0V \pm 5\%$ )

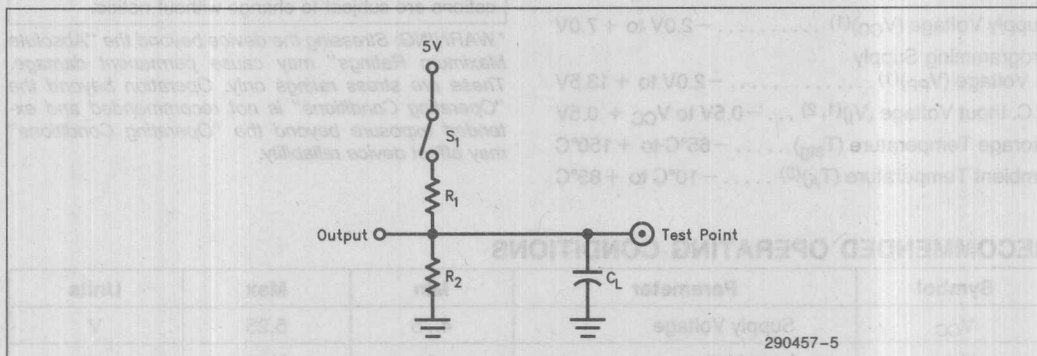
Symbol	Parameter	Min	Typ	Max	Units	Test Conditions
$V_{IH}^{(4)}$	High Level Input Voltage	2.0		$V_{CC} + 0.3$	V	
$V_{IL}^{(4)}$	Low Level Input Voltage	-0.3		0.8	V	
$V_{OH}$	TTL High Output Voltage	2.4			V	$I_O = -4 \text{ mA D.C.}, V_{CC} = \text{Min}$
	CMOS High Output Voltage	$V_{CC} - 0.3$			V	$I_O = -100 \text{ pA} = V_{CC} \text{ Min}$
$V_{OL}$	Low Level Output Voltage			0.45	V	$I_O = 16 \text{ mA D.C.}, V_{CC} = \text{Min}$
$I_I$	Input Leakage Current			10	$\mu\text{A}$	$V_{CC} = \text{Max.}, \text{GND} < V_{IN} < V_{CC}$
$I_{OZ}$	Output Leakage Current			10	$\mu\text{A}$	$V_{CC} = \text{Max.}, \text{GND} < V_{OUT} < V_{CC}$
$I_{SC}^{(5)}$	Output Short Circuit Current			120	mA	$V_{CC} = \text{Max.}, V_{OUT} = 0.5V$
$I_{CC}$	Power Supply Current (See $I_{CC}$ vs. Freq. Graph)		90	130	mA	$V_{CC} = \text{Max.}, V_{IN} = V_{CC}$ or GND, No Load, $f_{IN} = 15 \text{ MHz}$ , Device Prog. as a 10-Bit Counter

**NOTES:**

1. Voltages with respect to GND.
2. Minimum D.C. input is -0.5V. During transitions, the inputs may undershoot to -2.0V or overshoot to +7.0V for periods of less than 20 ns under no load conditions.
3. Under bias. Extended Temperature versions are also available.
4. Absolute values with respect to device GND; all over and undershoots due to system or tester noise are included.
5. Not more than 1 output should be tested at a time. Duration of that test should not exceed 1 second.

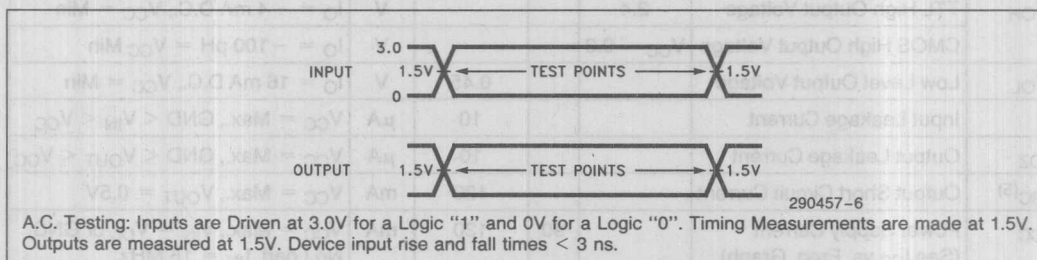


# A.C. TESTING LOAD CIRCUIT



Specification	S <sub>1</sub>	C <sub>L</sub>	Commercial		Measured Output Value
			R <sub>1</sub>	R <sub>2</sub>	
t <sub>PD</sub> , t <sub>CO</sub>	Closed	50 pF	240Ω	160Ω	1.5V
t <sub>PZX</sub>	Z → H: Open Z → L: Closed				1.5V
t <sub>PXZ</sub>	H → Z: Open L → Z: Closed	5 pF	240Ω	160Ω	H → Z: V <sub>OH</sub> - 0.5V L → Z: V <sub>OL</sub> + 0.5V

# A.C. TESTING INPUT, OUTPUT WAVEFORM



# CAPACITANCE (T<sub>A</sub> = 0°C to 70°C; V<sub>CC</sub> = 5.0V ± 5%)(6)

Symbol	Parameter	Min	Typ	Max	Units	Conditions
C <sub>IN</sub>	Input Capacitance		5	8	pF	V <sub>IN</sub> = 0V, f = 1.0 MHz
C <sub>IO</sub>	I/O Capacitance		6	8	pF	V <sub>OUT</sub> = 0V, f = 1.0 MHz
C <sub>CLK</sub>	CLK Capacitance		15	17	pF	V <sub>OUT</sub> = 0V, f = 1.0 MHz

**COMBINATORIAL MODE A.C. CHARACTERISTICS**(T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5.0V ± 5%)(7)

Symbol	Parameter	iPLD22V10-10			iPLD22V10-15			PLD22V10-25			Units
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
t <sub>PD</sub> <sup>(8)</sup>	Input or I/O to Output Valid—w/10 Outputs Switching	3		10	3		15			25	ns
t <sub>PZX</sub> <sup>(9)</sup>	Input or I/O to Output Enable	3		10	3		15			25	ns
t <sub>PXZ</sub> <sup>(9)</sup>	Input or I/O to Output Disable	3		10	3		15			25	ns
t <sub>CLR</sub>	Input or I/O to Asynch. Reset			15			20			20	ns

**NOTES:**

6. These values are evaluated at initial characterization and whenever design modifications occur that may affect capacitance.

7. Typical values are at T<sub>A</sub> = +25°C, V<sub>CC</sub> = 5V.

8. Ten outputs switching.

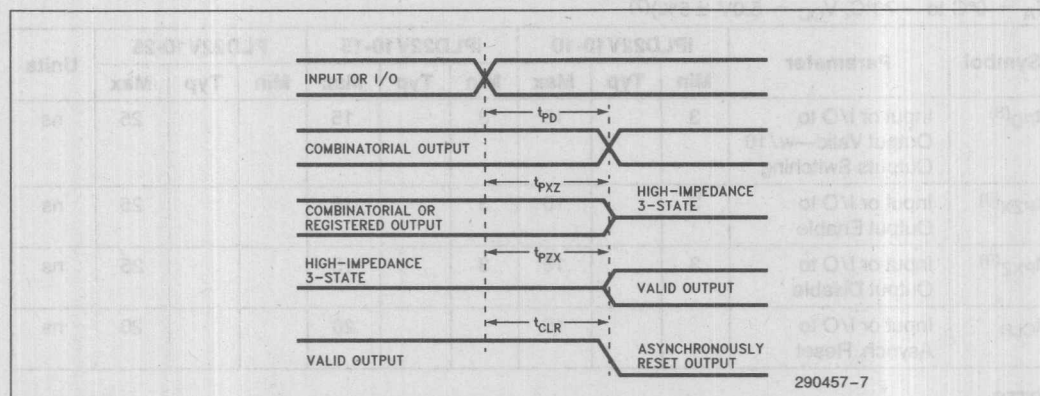
9. t<sub>PZX</sub> and t<sub>PXZ</sub> are measured at ±0.5V from steady state voltage as driven by spec. output load. t<sub>PXZ</sub> is measured with C<sub>L</sub> = 5 pF. Z → H and Z → L are measured at 1.5V on output.

10. Measured with device configured as a 10-bit counter.

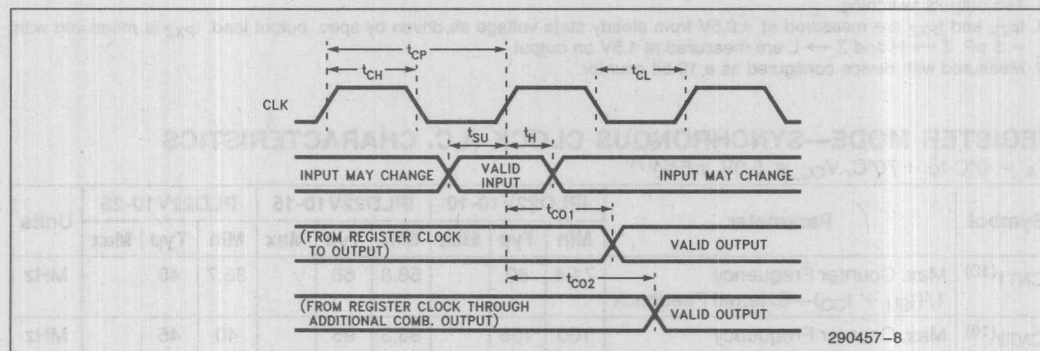
**REGISTER MODE—SYNCHRONOUS CLOCK A.C. CHARACTERISTICS**(T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5.0V ± 5%)(7)

Symbol	Parameter	iPLD22V10-10			iPLD22V10-15			PLD22V10-25			Units
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
f <sub>CNT1</sub> <sup>(10)</sup>	Max. Counter Frequency 1/(t <sub>SU</sub> + t <sub>CO</sub> )—External Feedback	71.4	80		58.8	66		35.7	40		MHz
f <sub>CNT2</sub> <sup>(10)</sup>	Max. Counter Frequency 1/(t <sub>CNT</sub> )—Internal Feedback	100	105		83.3	95		40	45		MHz
f <sub>MAX</sub>	Max. Frequency (Pipelined) 1/t <sub>CP</sub> —No Feedback	100	110		83.3	95		40	45		MHz
t <sub>SU</sub>	Input or I/O Setup Time to CLK or SP	7			9			14			ns
t <sub>H</sub>	Input or I/O Hold Time from CLK	0			0			0			ns
t <sub>CO1</sub>	CLK to Output Valid	3		7	2		8	2		14	ns
t <sub>CO2</sub>	CLK to Output Valid Fed Through Combinatorial Macrocell			16			18			30	ns
t <sub>CNT</sub>	Register Output Feedback to Register Input—Internal Path			10			12			25	ns
t <sub>CL</sub>	CLK Low Time	4			5			10			ns
t <sub>CH</sub>	CLK High Time	4			5			10			ns
t <sub>CP</sub>	CLK Period	10			12			25			ns
t <sub>arw</sub>	Asynchronous Reset Pulse Duration	4			5			5			ns
t <sub>arr</sub>	Asynchronous Reset to CLK ↑ Recovery Time	7			9			10			ns

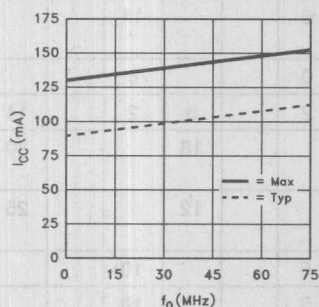
# COMBINATORIAL MODE



# SYNCHRONOUS REGISTERED MODE



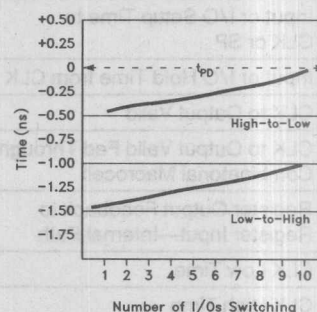
iPLD22V10  $I_{CC}$  vs Frequency



290457-9

Conditions:  
 $T_A = 0^\circ$   
 $V_{CC} = 5.25V$

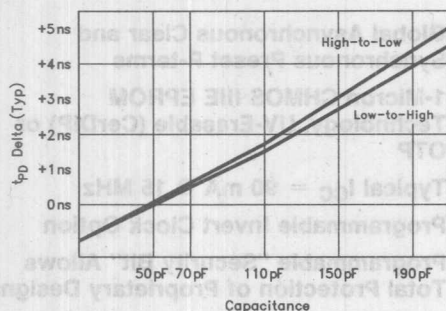
$t_{PD}$  Derating vs Number of Outputs Switching



290457-10

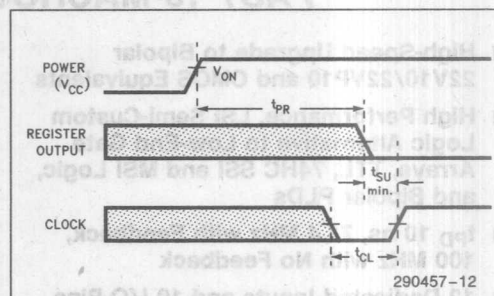
Conditions:  
 $T_A = 70^\circ C$   
 $V_{CC} = 4.75V$   
 $C_L = 50 pF$

**$t_{PD}$  Derating vs Capacitive Loading**



Conditions:  
 $T_A = 70^\circ\text{C}$   
 $V_{CC} = 4.75\text{V}$

## POWER-UP RESET



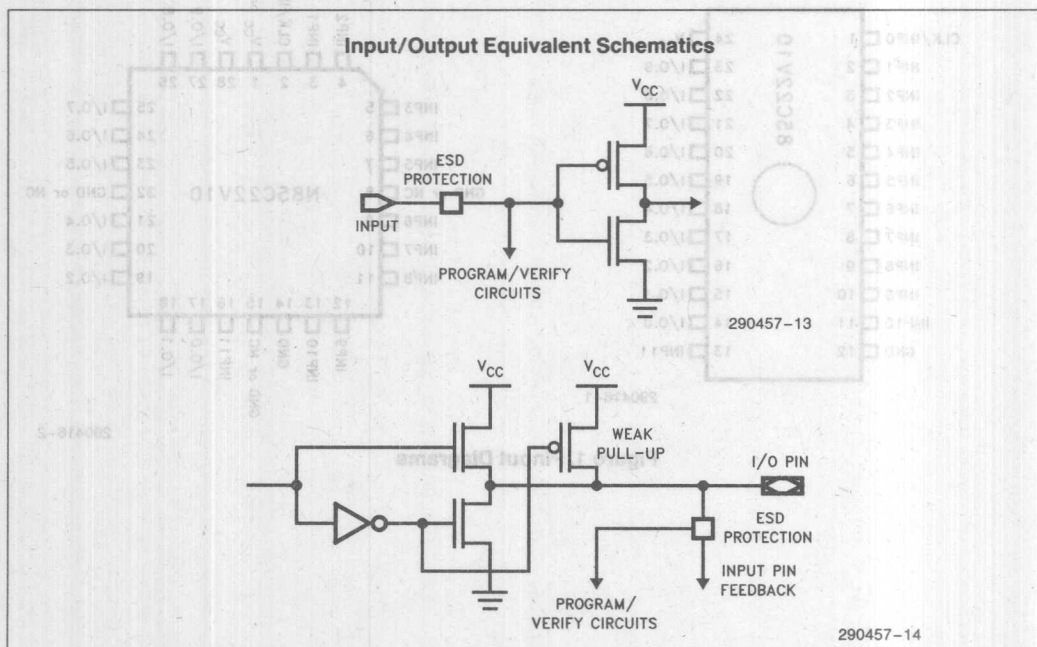
## POWER-UP RESET CHARACTERISTICS

Parameter Symbol	Parameter Description	Value
$t_{PR}$	Power-Up Reset Time	1000 ns Max.
$V_{ON}$	Turn-On Voltage	4.75V

## POWER-UP RESET

Internal power-up reset circuits ensure that all flip-flops will be reset to a logic 0 after the device has powered up. Because  $V_{CC}$  rise can vary significantly from one application to another,  $V_{CC}$  rise must be monotonic.

## Input/Output Equivalent Schematics



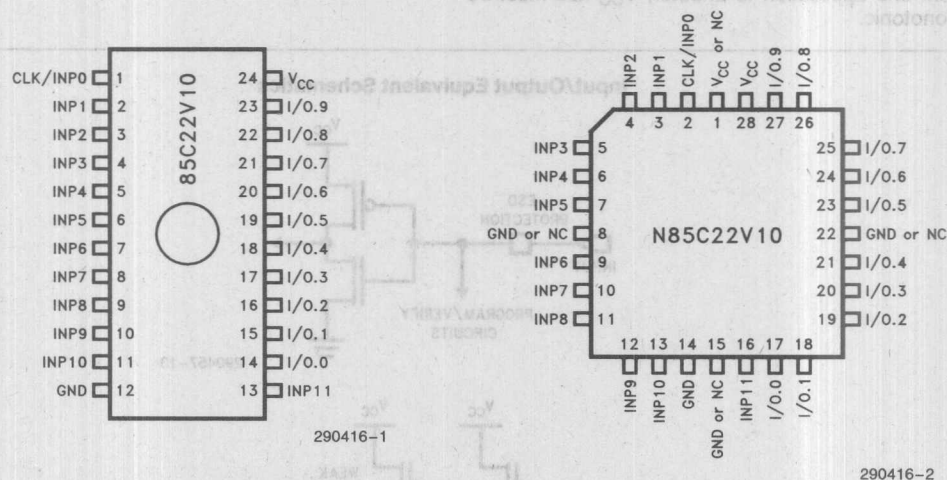


# 85C22V10

## FAST 10-MACROCELL CHMOS $\mu$ PLD

- High-Speed Upgrade to Bipolar 22V10/22VP10 and CMOS Equivalents
  - High Performance, LSI Semi-Custom Logic Alternative to Low-End Gate Arrays, TTL, 74HC SSI and MSI Logic, and Bipolar PLDs
  - $t_{PD}$  10 ns, 71.4 MHz with Feedback, 100 MHz with No Feedback
  - 12 Dedicated Inputs and 10 I/O Pins
  - 10 Macrocells with Programmable I/O Architecture (Register/Combinatorial)
  - Variable P-terms—Up to 16 per Macrocell, Selectable Output Polarity, Separate Output Enable P-term
  - Global Asynchronous Clear and Synchronous Preset P-terms
  - 1-Micron CHMOS IIIE EPROM Technology, UV-Erasable (CerDIP) or OTP
  - Typical  $I_{CC} = 90$  mA @ 15 MHz
  - Programmable Invert Clock Option
  - Programmable "Security Bit" Allows Total Protection of Proprietary Designs
  - 100% Generically Tested Logic Array
  - Available in 300-mil 24-Pin CerDIP/PDIP and 28-Pin PLCC Packages
- (See Packaging Spec., Order Number 240800, Package Type D, N and P)

Power-Up Reset Time	1000 ns Max.
Turn-On Voltage	4.75V
V <sub>ON</sub>	



## INTRODUCTION

The 85C22V10 is a high-performance, high-integration, general-purpose CMOS PLD. The 85C22V10 accommodates logic functions with up to 22 inputs and 10 I/O macrocells. I/O macrocells include an average of 12 p-terms for input, with a separate p-term for output enable. Figure 2 shows the global architecture of the device.

## JEDEC AND PIN COMPATIBILITY

The 85C22V10 is 100% JEDEC-, pin- and function-compatible with the industry-standard 22V10 PLD. JEDEC files developed for 22V10 devices can be used to program the 85C22V10. For designs requiring the 85C22V10 *superset* features, a new JEDEC must be developed. When the N85C22V10 (28-pin PLCC) is used to replace a conventional 22V10 in an existing design socket, pins 8, 15, 22 and 1 are left as No Connects (NC). New designs can take advantage of the additional device  $V_{CC}$  and grounds these pins offer.

## PROGRAMMABLE MACROCELLS

In addition to the 12 dedicated input pins, the 85C22V10 contains 10 programmable macrocells. Each of the macrocells can be programmed to function as an input or as a combinatorial or registered output. Programmable output polarity and program-

mable feedback options allow the 85C22V10 to be tailored to the precise needs of the target application. Figure 3 shows the architecture of each macrocell.

## Output Polarity

The output polarity for each 85C22V10 macrocell is programmable. Each combinatorial or registered output can be active-high or active-low.

## Feedback Options

85C22V10 macrocells programmed as combinatorial outputs support pin feedback to the logic array (i.e., feedback from the I/O pin). 85C22V10 macrocells programmed as registers allow internal register feedback to the logic array. These options are supported on both the 22V10 and 22VP10 devices.

85C22V10 macrocells programmed as registers also allow feedback to the logic array from the I/O pin. This feature is also supported on 22VP10 devices.

In addition, however, the 85C22V10 provides a *superset* feature with its ability to implement a combinatorial output with registered feedback to the logic array. This feedback option allows a single 85C22V10 macrocell to implement designs that would take up two macrocells in 22V10/22VP10 devices.

2

## ORDERING INFORMATION

f <sub>CNT1</sub> (MHz)	f <sub>MAX</sub> (MHz)	t <sub>PD</sub> (ns)	Order Code	Package	Operating Range
71.4	100	10	D85C22V10-10	*CerDIP	Commercial
			P85C22V10-10	PDIP	Commercial
			N85C22V10-10	PLCC	Commercial
58.8	83.3	15	D85C22V10-15	*CerDIP	Commercial
			P85C22V10-15	PDIP	Commercial
			N85C22V10-15	PLCC	Commercial

\*Only the windowed CerDIP package allows UV erase.

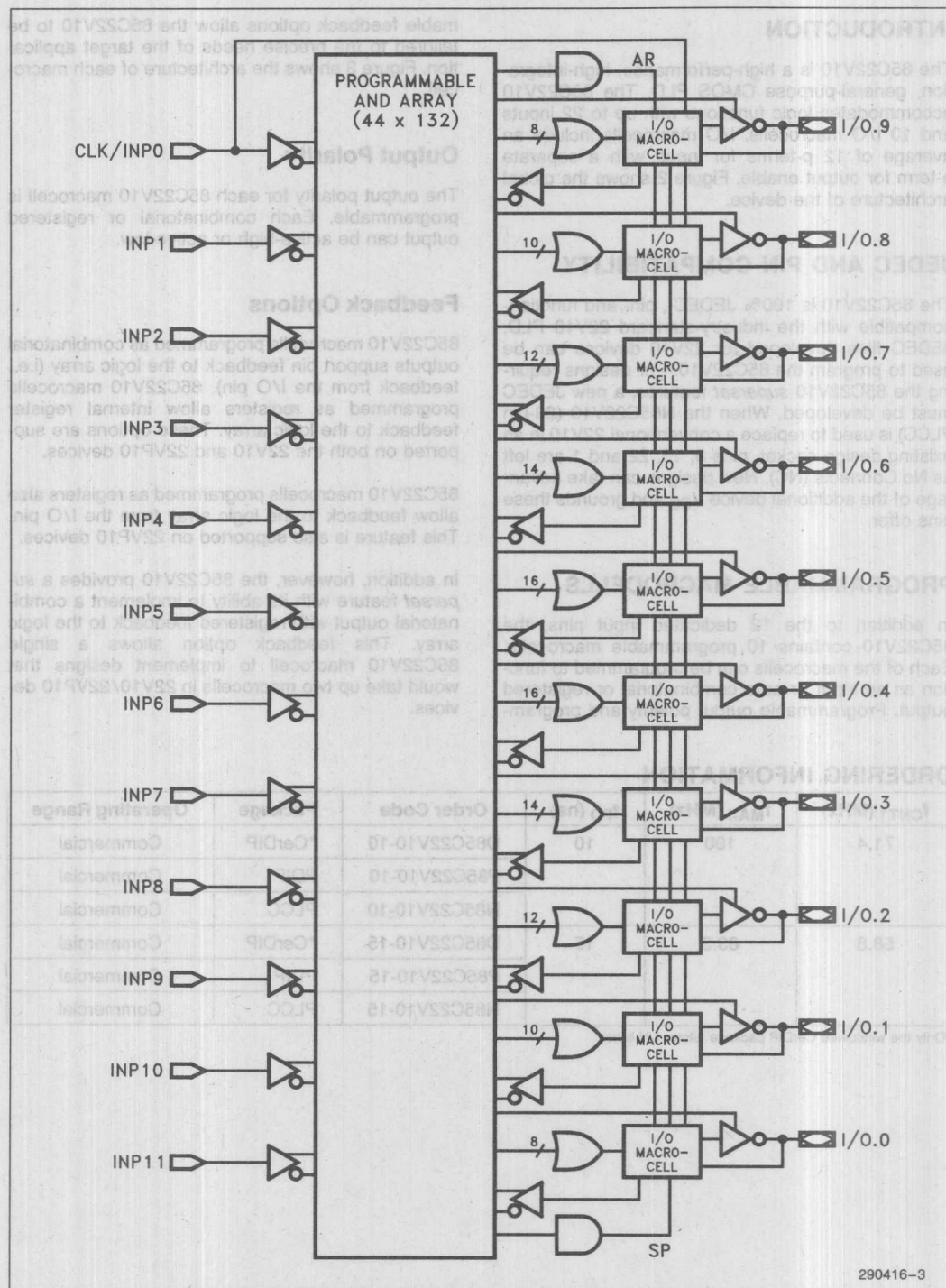


Figure 2. 85C22V10 Global Architecture

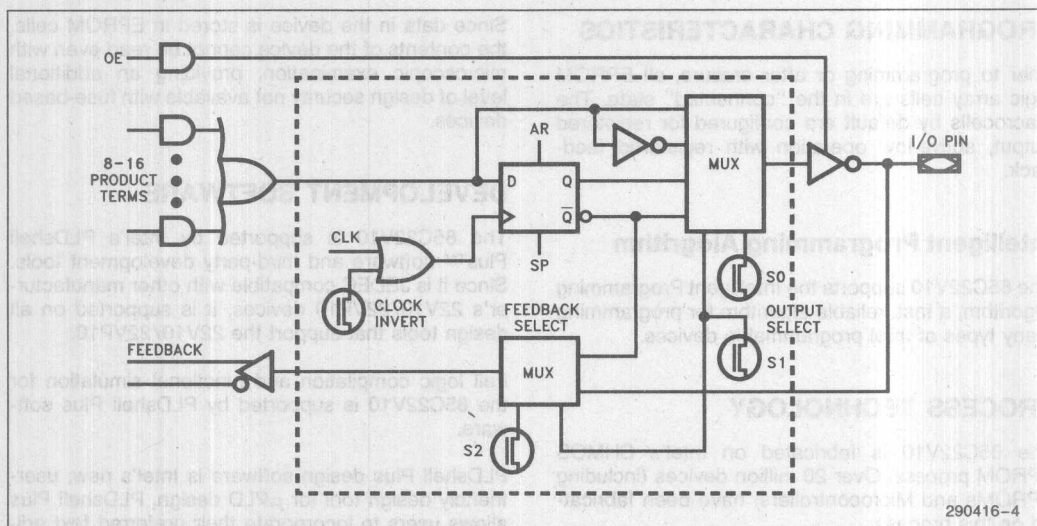


Figure 3. 85C22V10 Macrocell Architecture

Table 1 lists the macrocell configurations:

Table 1. 85C22V10 Macrocell Configurations

S2	S1	S0	Output/Polarity	Feedback
0	0	0	Registered/Active Low	Registered
0	0	1	Registered/Active High	Registered
0	1	0	Combinatorial/Active Low	Pin
0	1	1	Combinatorial/Active High	Pin
1	0	0	**Registered/Active Low	Pin
1	0	1	**Registered/Active High	Pin
1	1	0	*Combinatorial/Active Low	Registered
1	1	1	*Combinatorial/Active High	Registered

\*Not available on the 22V10 or 22VP10.

\*\*Not available on the 22V10.

### Clock Invert

A clock invert option for each macrocell allows macrocell registers to be independently clocked on the rising or falling edge of the global clock. This *super-set* feature allows the 85C22V10 to implement designs that could not be implemented in a 22V10/22VP10 device.

### Register Preset/Reset

85C22V10 macrocell registers can be preset or reset using global preset and reset p-terms. Register preset is synchronous and must meet the specified setup time to the clock signal. Register reset is asynchronous and has no setup requirement to the clock. Preset and reset set or reset the register. Output polarity is selected separately.

### Programmable Output Enable

Each macrocell contains an output buffer that can place the respective output in a high-impedance state (three-state). The output buffer is controlled by a single p-term per macrocell in the logic array and is asynchronous.

### POWER-ON CHARACTERISTICS

85C22V10 inputs and outputs begin responding  $1 \mu\text{s}$  (max.) after  $V_{CC}$  power-up ( $V_{CC} = 4.75\text{V}$ ) or after a power-loss/power-up sequence. All macrocells programmed as registers are set to a logic low.

### ERASURE CHARACTERISTICS

Erase time for the 85C22V10 is  $2\frac{1}{2}$  hours at  $12,000 \mu\text{W}/\text{cm}^2$  with a  $2537\text{\AA}$  lamp.

Erase begins upon exposure to light with wavelengths shorter than approximately  $4000\text{\AA}$ . Sunlight and certain types of fluorescent lamps have wavelengths in the  $3000\text{\AA}$  to  $4000\text{\AA}$  range. Erase data indicates that constant exposure to room level fluorescent lighting will erase the device in approximately six years. It would take approximately two weeks of constant exposure to direct sunlight to erase the device.



## PROGRAMMING CHARACTERISTICS

Prior to programming or after erasure, all EPROM logic array cells are in the "connected" state. The macrocells by default are configured for registered output, active-low operation with registered feedback.

### Intelligent Programming Algorithm

The 85C22V10 supports the Intelligent Programming Algorithm, a fast, reliable algorithm for programming many types of Intel programmable devices.

## PROCESS TECHNOLOGY

The 85C22V10 is fabricated on Intel's CHMOS EPROM process. Over 20 million devices (including EPROMs and Microcontrollers) have been fabricated on this process.

## TESTABILITY

The 85C22V10 is completely tested at the factory. Unlike fuse-based PLDs, which have one-time programmable fuse links that limit testing to small-scale sampling, each EPROM cell in the 85C22V10 is tested and erased prior to shipment.

## REGISTER PRELOAD

85C22V10 macrocell registers can be preloaded with any pattern to allow testing of all possible logic states. Information on register preload for test purposes is available from Intel.

## SECURITY

A single programmable bit, called the security bit or verify protect bit, controls access to the data programmed into the device. Once this security bit is set, the design cannot be copied. The security bit is cleared via UV-eraser along with device contents.

Since data in the device is stored in EPROM cells, the contents of the device cannot be read even with microscopic examination, providing an additional level of design security not available with fuse-based devices.

## DEVELOPMENT SOFTWARE

The 85C22V10 is supported by Intel's PLDshell Plus™ software and third-party development tools. Since it is JEDEC compatible with other manufacturer's 22V10/22VP10 devices, it is supported on all design tools that support the 22V10/22VP10.

Full logic compilation and functional simulation for the 85C22V10 is supported by PLDshell Plus software.

PLDshell Plus design software is Intel's new, user-friendly design tool for  $\mu$ PLD design. PLDshell Plus allows users to incorporate their preferred text editor, programming software, and additional design tools into an easy-to-use, menued design environment that includes Intel's PLDasm™ logic compiler and simulation software along with disassembly, conversion, and translation utilities. The PLDasm compiler and simulator software accepts industry-standard PDS source files that express designs as Boolean equations, truth tables, or state machines. On-line help, datasheet briefs, technical notes, and error message information, along with waveform viewing/printing capability make the design task as easy as possible. PLDshell Plus software is available from Intel Literature channels or from your local Intel sales representative, order #468810.

Tools that support schematic capture and timing simulation for the 85C22V10 are available. Support under iPLS II is still available. Please refer to the "Development Tools" section of the Programmable Logic handbook.

**ABSOLUTE MAXIMUM RATINGS\***

Supply Voltage ( $V_{CC}$ )<sup>(1)</sup> ..... -2.0V to +7.0V  
 Programming Supply  
 Voltage ( $V_{PP}$ )<sup>(1)</sup> ..... -2.0V to +13.5V  
 D.C. Input Voltage ( $V_I$ )<sup>(1, 2)</sup> ... -0.5V to  $V_{CC} + 0.5V$   
 Storage Temperature ( $T_{stg}$ ) ..... -65°C to +150°C  
 Ambient Temperature ( $T_A$ )<sup>(3)</sup> ..... -10°C to +85°C

NOTICE: This data sheet contains preliminary information on new products in production. The specifications are subject to change without notice. Verify with your local Intel Sales office that you have the latest data sheet before finalizing a design.

*\*WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

**RECOMMENDED OPERATING CONDITIONS**

Symbol	Parameter	Min	Max	Units
$V_{CC}$	Supply Voltage	4.75	5.25	V
$V_{IN}$	Input Voltage	0	$V_{CC}$	V
$V_O$	Output Voltage	0	$V_{CC}$	V
$T_A$	Operating Temperature	0	+70	°C
$t_R$	Input Rise Time		500	ns
$t_F$	Input Fall Time		500	ns

2

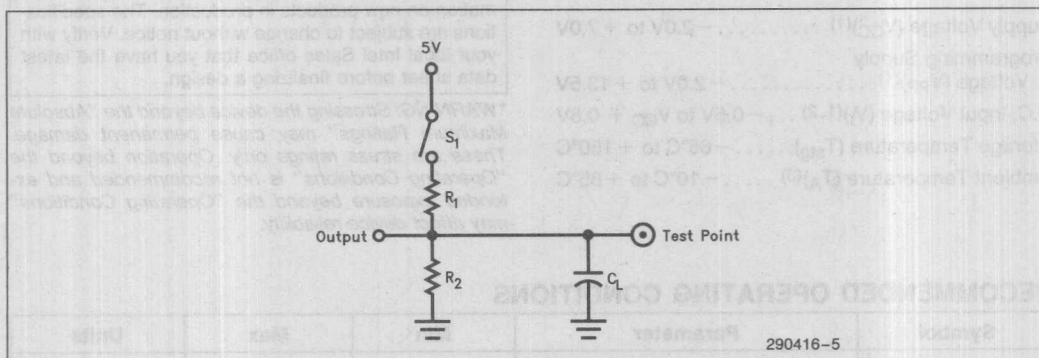
**D.C. CHARACTERISTICS** ( $T_A = 0^\circ\text{C}$  to  $+70^\circ\text{C}$ ,  $V_{CC} = 5.0V \pm 5\%$ )

Symbol	Parameter	Min	Typ	Max	Units	Test Conditions
$V_{IH}^{(4)}$	High Level Input Voltage	2.0		$V_{CC} + 0.3$	V	
$V_{IL}^{(4)}$	Low Level Input Voltage	-0.3		0.8	V	
$V_{OH}$	TTL High Output Voltage	2.4			V	$I_O = -4 \text{ mA D.C.}, V_{CC} = \text{Min}$
	CMOS High Output Voltage	$V_{CC} - 0.3$			V	$I_O = -100 \text{ pA}, V_{CC} = \text{Min}$
$V_{OL}$	Low Level Output Voltage			0.45	V	$I_O = 16 \text{ mA D.C.}, V_{CC} = \text{Min}$
$I_I$	Input Leakage Current			10	$\mu\text{A}$	$V_{CC} = \text{Max.}, \text{GND} < V_{IN} < V_{CC}$
$I_{OZ}$	Output Leakage Current			10	$\mu\text{A}$	$V_{CC} = \text{Max.}, \text{GND} < V_{OUT} < V_{CC}$
$I_{SC}^{(5)}$	Output Short Circuit Current			120	mA	$V_{CC} = \text{Max.}, V_{OUT} = 0.5V$
$I_{CC}$	Power Supply Current (See $I_{CC}$ vs. Freq. Graph)		90	130	mA	$V_{CC} = \text{Max.}, V_{IN} = V_{CC}$ or GND, No Load, $f_{IN} = 15 \text{ MHz}$ , Device Prog. as a 10-Bit Counter

**NOTES:**

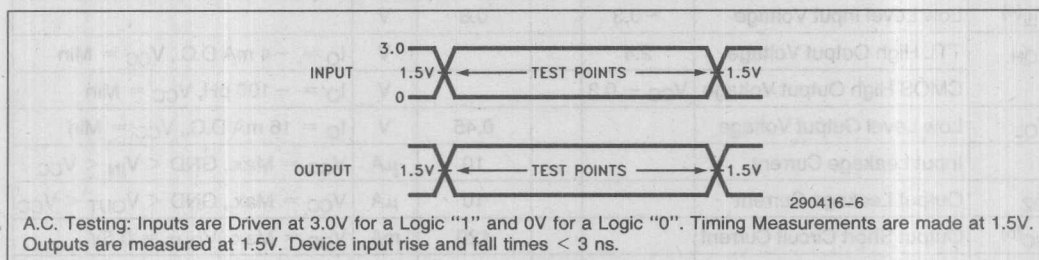
1. Voltages with respect to GND.
2. Minimum D.C. input is -0.5V. During transitions, the inputs may undershoot to -2.0V or overshoot to +7.0V for periods of less than 20 ns under no load conditions.
3. Under bias. Extended Temperature versions are also available.
4. Absolute values with respect to device GND; all over and undershoots due to system or tester noise are included.
5. Not more than 1 output should be tested at a time. Duration of that test should not exceed 1 second.

## A.C. TESTING LOAD CIRCUIT



Specification	S <sub>1</sub>	C <sub>L</sub>	Commercial		Measured Output Value
			R <sub>1</sub>	R <sub>2</sub>	
t <sub>PD</sub> , t <sub>CO</sub>	Closed	50 pF	240Ω	160Ω	1.5V
t <sub>PZX</sub>	Z → H: Open Z → L: Closed				1.5V
t <sub>pxz</sub>	H → Z: Open L → Z: Closed	5 pF			H → Z: V <sub>OH</sub> - 0.5V L → Z: V <sub>OL</sub> + 0.5V

## A.C. TESTING INPUT, OUTPUT WAVEFORM

CAPACITANCE (T<sub>A</sub> = 0°C to 70°C; V<sub>CC</sub> = 5.0V ± 5%)(6)

Symbol	Parameter	Min	Typ	Max	Units	Conditions
C <sub>IN</sub>	Input Capacitance		5	8	pF	V <sub>IN</sub> = 0V, f = 1.0 MHz
C <sub>IO</sub>	I/O Capacitance		6	8	pF	V <sub>OUT</sub> = 0V, f = 1.0 MHz
C <sub>CLK</sub>	CLK Capacitance		15	17	pF	V <sub>OUT</sub> = 0V, f = 1.0 MHz

**COMBINATORIAL MODE A.C. CHARACTERISTICS**(T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5.0V ± 5%)(7)

Symbol	Parameter	85C22V10-10			85C22V10-15			Units
		Min	Typ	Max	Min	Typ	Max	
t <sub>PD</sub> <sup>(8)</sup>	Input or I/O to Output Valid—w/10 Outputs Switching	3		10	3		15	ns
t <sub>PZX</sub> <sup>(9)</sup>	Input or I/O to Output Enable	3		10	3		15	ns
t <sub>PXZ</sub> <sup>(9)</sup>	Input or I/O to Output Disable	3		10	3		15	ns
t <sub>CLR</sub>	Input or I/O to Asynch. Reset			15			20	ns

**NOTES:**

6. These values are evaluated at initial characterization and whenever design modifications occur that may affect capacitance.

7. Typical values are at T<sub>A</sub> = +25°C, V<sub>CC</sub> = 5V.

8. Ten outputs switching.

9. t<sub>PZX</sub> and t<sub>PXZ</sub> are measured at ±0.5V from steady state voltage as driven by spec. output load. t<sub>PXZ</sub> is measured with C<sub>L</sub> = 5 pF. Z → H and Z → L are measured at 1.5V on output.

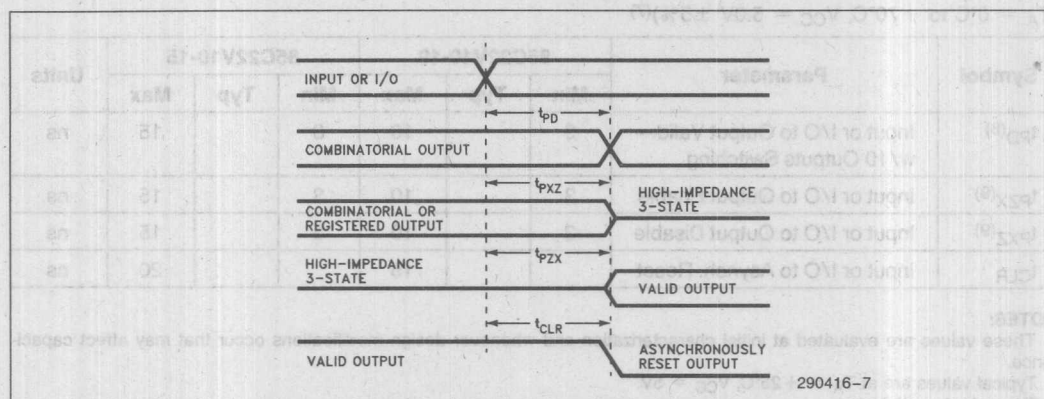
2

**REGISTER MODE—SYNCHRONOUS CLOCK A.C. CHARACTERISTICS**(T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5.0V ± 5%)(7)

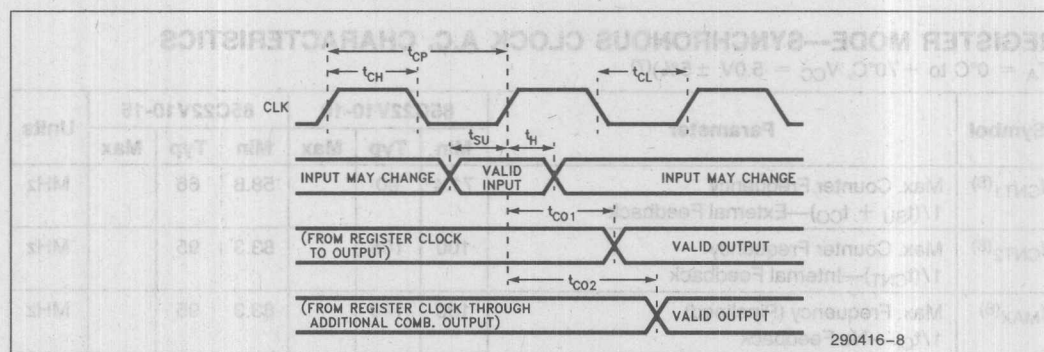
Symbol	Parameter	85C22V10-10			85C22V10-15			Units
		Min	Typ	Max	Min	Typ	Max	
f <sub>CNT1</sub> <sup>(8)</sup>	Max. Counter Frequency 1/(t <sub>SU</sub> + t <sub>CO</sub> )—External Feedback	71.4	80		58.8	66		MHz
f <sub>CNT2</sub> <sup>(8)</sup>	Max. Counter Frequency 1/t <sub>CNT</sub> —Internal Feedback	100	105		83.3	95		MHz
f <sub>MAX</sub> <sup>(8)</sup>	Max. Frequency (Pipelined) 1/t <sub>CP</sub> —No Feedback	100	110		83.3	95		MHz
t <sub>SU</sub>	Input or I/O Setup Time to CLK or SP	7			9			ns
t <sub>SU</sub>	Input or I/O Setup Time to Inverted CLK	7			9			ns
t <sub>H</sub>	Input or I/O Hold Time from CLK	0			0			ns
t <sub>CO1</sub>	CLK to Output Valid	3		7	2		8	ns
t <sub>CO1</sub>	Inverted CLK to Output Valid	3		7	2		8	ns
t <sub>CO2</sub>	CLK to Output Valid Fed Through Combinatorial Macrocell			16			18	ns
t <sub>CNT</sub>	Register Output Feedback to Register Input—Internal Path			10			12	ns
t <sub>CL</sub>	CLK Low Time	4			5			ns
t <sub>CH</sub>	CLK High Time	4			5			ns
t <sub>CP</sub>	CLK Period	10			12			ns
t <sub>arw</sub>	Asynchronous Reset Pulse Duration	4			5			ns
t <sub>arr</sub>	Asynchronous Reset to CLK ↑ Recovery Time	7			9			ns



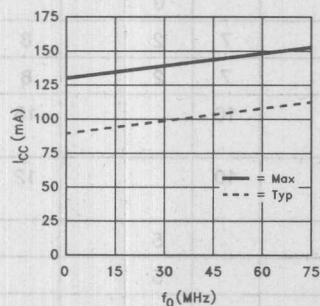
# COMBINATORIAL MODE



# SYNCHRONOUS REGISTERED MODE



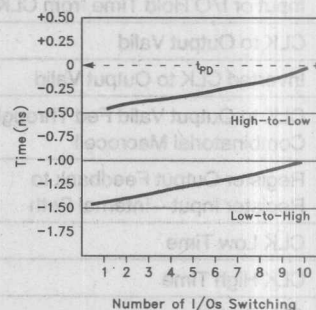
85C22V10  $I_{CC}$  vs Frequency



290416-9

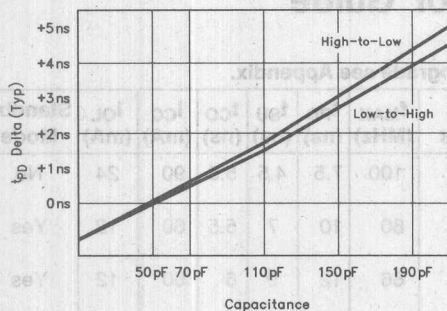
Conditions:  
 $T_A = 0^\circ$   
 $V_{CC} = 5.25V$

$t_{PD}$  Derating vs Number of Outputs Switching



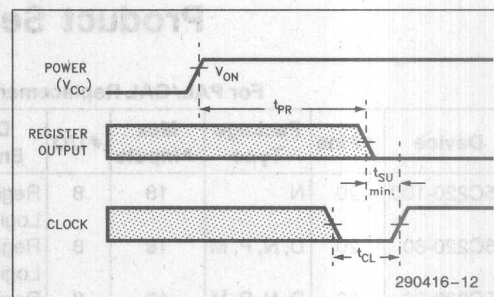
290416-10

Conditions:  
 $T_A = 70^\circ C$   
 $V_{CC} = 4.75V$   
 $C_L = 50 pF$

**$t_{PD}$  Derating vs Capacitive Loading**

290416-11

Conditions:  
 $T_A = 70^\circ\text{C}$   
 $V_{CC} = 4.75\text{V}$

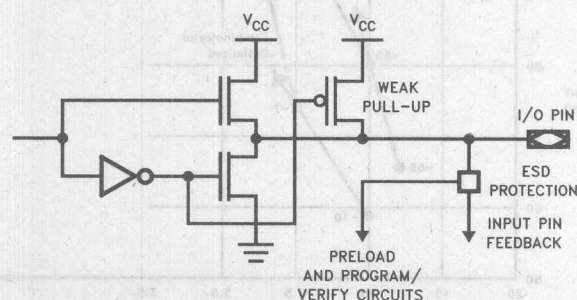
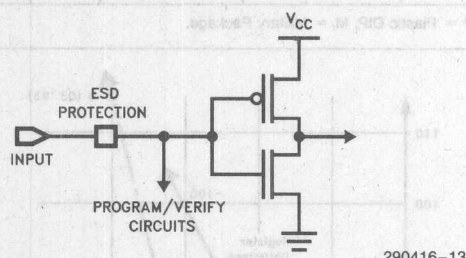
**POWER-UP RESET****POWER-UP RESET CHARACTERISTICS**

Parameter Symbol	Parameter Description	Value
$t_{PR}$	Power-Up Reset Time	1000 ns Max.
$V_{ON}$	Turn-On Voltage	4.75V

2

**POWER-UP RESET**

Internal power-up reset circuits ensure that all flip-flops will be reset to a logic 0 after the device has powered up. Because  $V_{CC}$  rise can vary significantly from one application to another,  $V_{CC}$  rise must be monotonic.

**Input/Output Equivalent Schematics**

290416-14

# 85C220/85C224 Family Product Selector Guide

For PAL/GAL Replacement or Upgrade see Appendix.

Device	# Pins	Package Type	Max # Inputs	# I/O	Design Emphasis	f <sub>MAX</sub> (MHz)	t <sub>PD</sub> (ns)	t <sub>SU</sub> (ns)	t <sub>CO</sub> (ns)	I <sub>CC</sub> (mA)	I <sub>OL</sub> (mA)	Standby Mode
85C220-100	20	N	18	8	Registered Logic	100	7.5	4.5	5.5	90	24	No
85C220-80	20	D, N, P, M	18	8	Registered Logic	80	10	7	5.5	60	12	Yes
85C220-66	20	D, N, P, M	18	8	Registered Logic	66	12	9	6	60	12	Yes
85C220-7	20	N, P	18	8	Combinatorial Logic	74	7.5	7	6.5	105	24	No
85C220-10	20	N, P	18	8	Combinatorial Logic	58.8	10	10	7	105	24	No
85C224-100	24	N	22	8	Registered Logic	100	7.5	4.5	5.5	90	24	No
85C224-80	24	D, N, P, M	22	8	Registered Logic	80	10	7	5.5	60	12	Yes
85C224-66	24	D, N, P, M	22	8	Registered Logic	66	12	9	6	60	12	Yes
85C224-7	24	N, P	22	8	Combinatorial Logic	74	7.5	7	6.5	105	24	No
85C224-10	24	N, P	22	8	Combinatorial Logic	58.8	10	10	7	105	24	No

D = Ceramic, DIP, N = PLCC, P = Plastic DIP, M = Military Package.

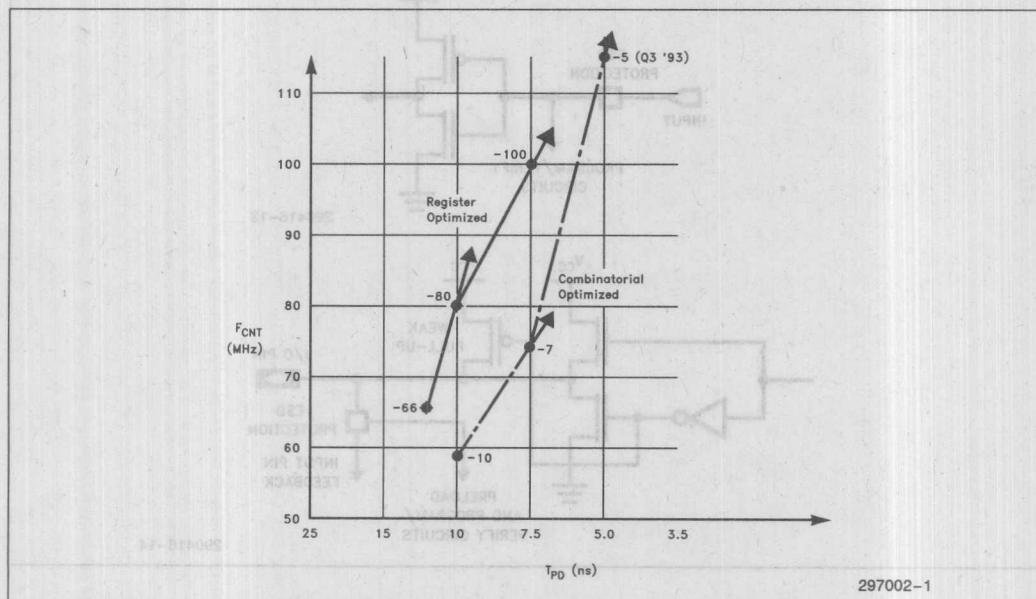


FIGURE 1. 85C220/85C224 Family Speed Bin Comparison



## 85C220/85C224-100, -80 AND -66 REGISTER OPTIMIZED TIMING FAST 1-MICRON CHMOS 8-MACROCELL $\mu$ PLDs

These register optimized timing  $\mu$ PLDs offer superior design features:

- Low-Power, High-Performance Upgrade for SSI/MSI Logic and Bipolar PALs/GALs in Advanced Intel Microprocessor i486™, i386™, i860™, 80960 Series and other High-Performance Systems
- Replacement or Upgrade for 16V8/20V8 PAL and GAL Architecture
- 8 P-Terms, Selectable SOP Invert, OE P-Term for Each Macrocell
- 8 Macrocells with Independently Programmable I/O Architecture (Register/Combinatorial)
- Up to 18 Inputs (10 Dedicated and 8 I/O) and 8 Outputs
- Programmable "Security Bit" Allows Total Protection of Proprietary Designs
- 100% Generally Tested Logic Array

### 85C220-100 AND 85C224-100

- 100 MHz Max Frequency (External Feedback); 5.5 ns (Max) Clock to Output; 4.5 ns (Min) Set-Up Time
- Meets Critical Timing Requirements of Advanced Intel Microprocessor Systems
- 7 ns (Max) Propagation Delay
- Typical  $I_{CC} = 90$  mA
- Available in 20-Pin and 28-Pin PLCC Packages

### 85C220-80 AND 85C224-80

- Extremely Low Power ( $I_{CC} = 40$  mA); Programmable "Standby" Option (25  $\mu$ A Typical)
- 80 MHz Max Frequency (External Feedback); 5.5 ns (Max) Clock to Output; 7 ns (Min) Set-Up Time
- Performance/Power Upgrade to "D- and E-Speed" PLDs in State Machine Applications
- 10 ns (Max) Propagation Delay, 100 MHz Max Frequency (Internal Feedback), 111 MHz (Pipelined)
- High-Speed Upgrade to EP320, EP330, and 5C032
- Available in 300-mil 20-Pin and 24-Pin CerDIP/PDIP Packages, and 20-Pin and 28-Pin PLCC Packages

2

Intel386™, i486™ and i860™ are trademarks of Intel Corporation.  
\*PAL is a registered trademark of Advanced Micro Devices.



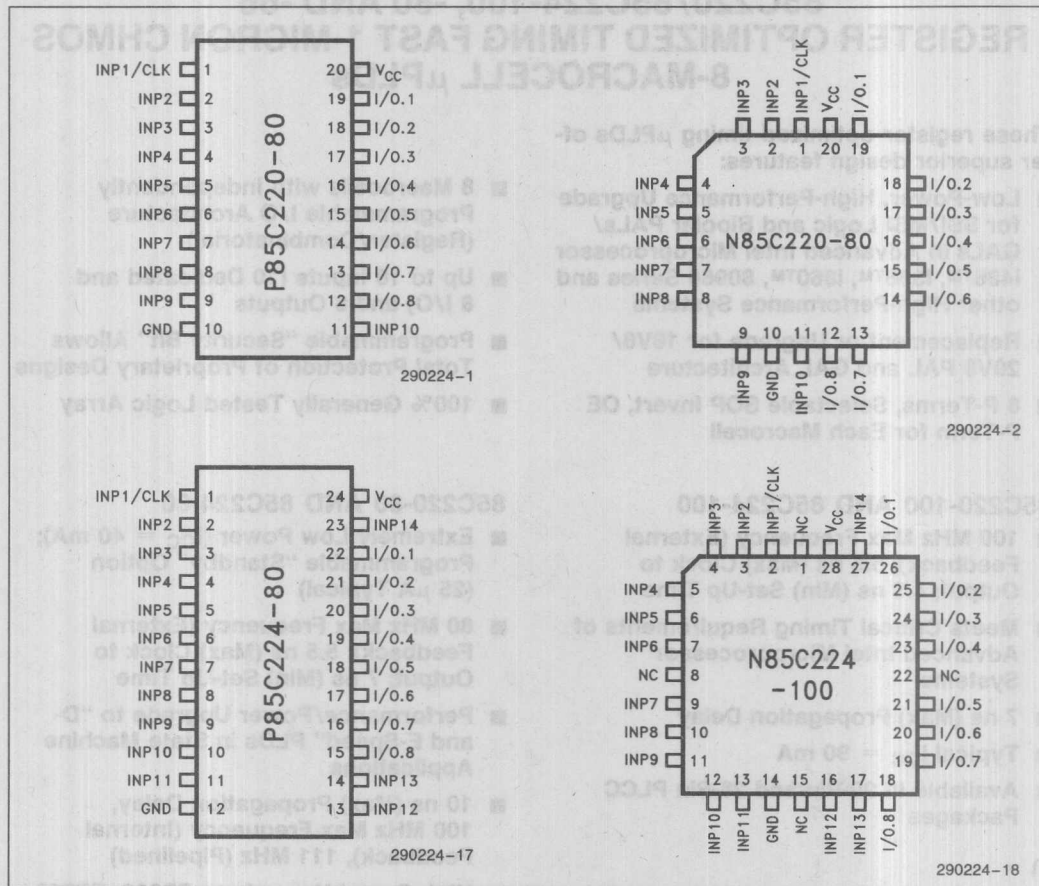


Figure 1. Pinout Diagrams

## INTRODUCTION

The Intel 85C220/85C224 8-micron CHMOS  $\mu$ PLD (Microcomputer Programmable Logic Device) is capable of implementing over 300 equivalent gates of user-customized logic functions through programming. With its flexible I/O architecture and fast speeds, this device has functional capabilities that surpass those of typical programmable logic devices. This device can be used to upgrade high-speed bipolar programmable logic devices and 74-series LS and CMOS SSI and MSI logic devices in bus control and state-machine applications for Advanced Intel Microprocessors, i486™, Intel386™, and Intel i860™-based systems and other high-performance processors. The 85C220/85C224 can also be used as a direct, low-power replacement for almost all high-speed 20-pin and 24-pin fuse-based programmable logic devices.

The 85C220/85C224 uses advanced EPROM cells as architecture and logic control memory elements. Coupled with Intel's proprietary CHMOS IIIE technology, these devices offer a fast  $t_{PD}$  in combinatorial mode, with current consumption much lower than bipolar devices of equivalent speed. The maximum "count" frequencies of 100 MHz and 80 MHz are optimized for high-performance state machines typically encountered in bus control applications. EPROM technology allows these devices to be 100% factory tested by programming and erasing all the EPROM logic control elements.

The inherent speed of the device together with its lower power demands and plastic package make the 85C220/85C224 an ideal production vehicle for high-volume manufacturing of high-performance systems. The 85C220/85C224 will improve performance and reliability, while decreasing system noise, power consumption and heat generation.

## ARCHITECTURE DESCRIPTION

The architecture of the 85C220/85C224 is based on the SOP (Sum of Products) PAL structure with a programmable AND array feeding into a fixed OR array. Programmable macrocells allow the device to accommodate both combinatorial and sequential logic functions. Each macrocell is individually programmable for combinatorial or registered output. An invert option on the SOP allows each output to be configured as an active-high or active-low output.

As shown in Figures 2 and 3, the 85C220/85C224 contains 10/14 dedicated inputs and 8 I/O pins. Each I/O pin can be individually programmed to function as an input, output, or bidirectional I/O pin. Associated with each I/O pin is a programmable macrocell.

Figures 4 and 5 show the structure of the 85C220/85C224 macrocell. Each macrocell includes a p-term (product term) block with eight AND p-terms feeding the OR gate of the I/O control block and one additional p-term controlling the output buffer. The logic array is 36 rows wide, allowing each p-term in the device to connect to the true or complement of each input and I/O feedback signal. Each intersecting point in the logic array is connected or not connected based on the value programmed in the EPROM array. Initially (EPROM erased state), all p-terms are connected to all signals. Connections are broken by programming the appropriate EPROM cells. Connecting both the true and complement of a signal for a given p-term removes that p-term from the SOP for the macrocell (i.e., that p-term is a "don't care").

Figure 6 shows the architecture of each macrocell's I/O control block. The SOP input to the I/O control block can be inverted or non-inverted. The output can be registered or combinatorial. When registered output is selected, feedback to the logic array comes directly from the register (before the output buffer). When combinatorial output is selected, feedback comes from the I/O pin (after the output buffer) and can be used for bidirectional I/O. The register is a D-type register that clocks on the rising edge of CLK.

## 20-PIN AND 24-PIN PLD COMPATIBILITY

The 85C220/85C224 is designed to be a logical superset of most high-speed 20-pin and 24-pin bipolar PAL and GAL devices. The I/O and logic sections of the device can be configured to emulate any of the devices listed below. Designers can often replace multiple PALs with fewer 85C220/85C224 devices. Tables 1 and 2 include some of the devices with which the 85C220/85C224 are compatible.

Table 1. Replacement/Upgrade

10 ns—20-Pin and 24-Pin		
Company	20-Pin Part	24-Pin Part
Intel	85C220-80	85C224-80
AMD	PAL16L8D	PAL20L8-10
AMD	PAL16R8D	PAL20R8-10
AMD	PAL16R4D	PAL20R4-10
AMD	PAL16R6D	PAL20R6-10
AMD	PAL16R8-7	PAL20R8-7
AMD	PALCE16V8	PALCE20V8
National	GAL16V8A	GAL20V8A
National	PAL16L8D	PAL20L8D
National	PAL16R4D	PAL20R4D
National	PAL16R6D	PAL20R6D
National	PAL16R8D	PAL20R8D
National	PAL16R8-7	N/A
Signetics	PLUS16L8D	PLUS20L8D
Signetics	PLUS16R4D	PLUS20R4D
Signetics	PLUS16R6D	PLUS20R6D
Signetics	PLUS16R8D	PLUS20R8D
Signetics	PLUS16R8-7	PLUS20R8-7
TI	TIBPAL16L8-10	TIBPAL20L8-10
TI	TIBPAL16R4-10	TIBPAL20R4-10
TI	TIBPAL16R6-10	TIBPAL20R6-10
TI	TIBPAL16R8-10	TIBPAL20R8-10
TI	TIBPAL16R8-7	TIBPAL20R8-7

Table 2. Replacement/Upgrade

12 ns—20-Pin and 24-Pin		
Company	20-Pin Part	24-Pin Part
Intel	85C220-66	85C224-66
AMD	PAL16L8	PAL20L8
AMD	PAL16R8	PAL20R8
AMD	PAL16R4	PAL20R4
AMD	PAL16R6	PAL20R6
AMD	PALCE16V8	PALCE20V8
Cypress	PALC16L8	PALC20L8
Cypress	PALC16R4	PALC20R4
Cypress	PALC16R6	PALC20R6
Cypress	PALC16R8	PALC20R8
Cypress	PLDC18G8	PLDC20G10
National	GAL16V8A	GAL20V8A
National	PAL10H8	N/A
National	PAL10L8	PAL14L8
National	PAL12H6	N/A
National	PAL12L6	PAL12L10
National	PAL14H4	N/A
National	PAL14L4	PAL18L4
National	PAL16C1	PAL20C1
National	PAL16H2	N/A
National	PAL16L2	PAL20L2
National	PAL16L6	N/A
National	PAL16L8	PAL20L8
National	PAL16R4	PAL20R4
National	PAL16R6	PAL20R6
National	PAL16R8	PAL20R8
Signetics	PLUS16L8	PLUS20L8
Signetics	PLUS16R4	PLUS20R4
Signetics	PLUS16R6	PLUS20R6
Signetics	PLUS16R8	PLUS20R8
TI	TIBPAL16L8	TIBPAL20L8
TI	TIBPAL16R4	TIBPAL20R4
TI	TIBPAL16R6	TIBPAL20R6
TI	TIBPAL16R8	TIBPAL20R8

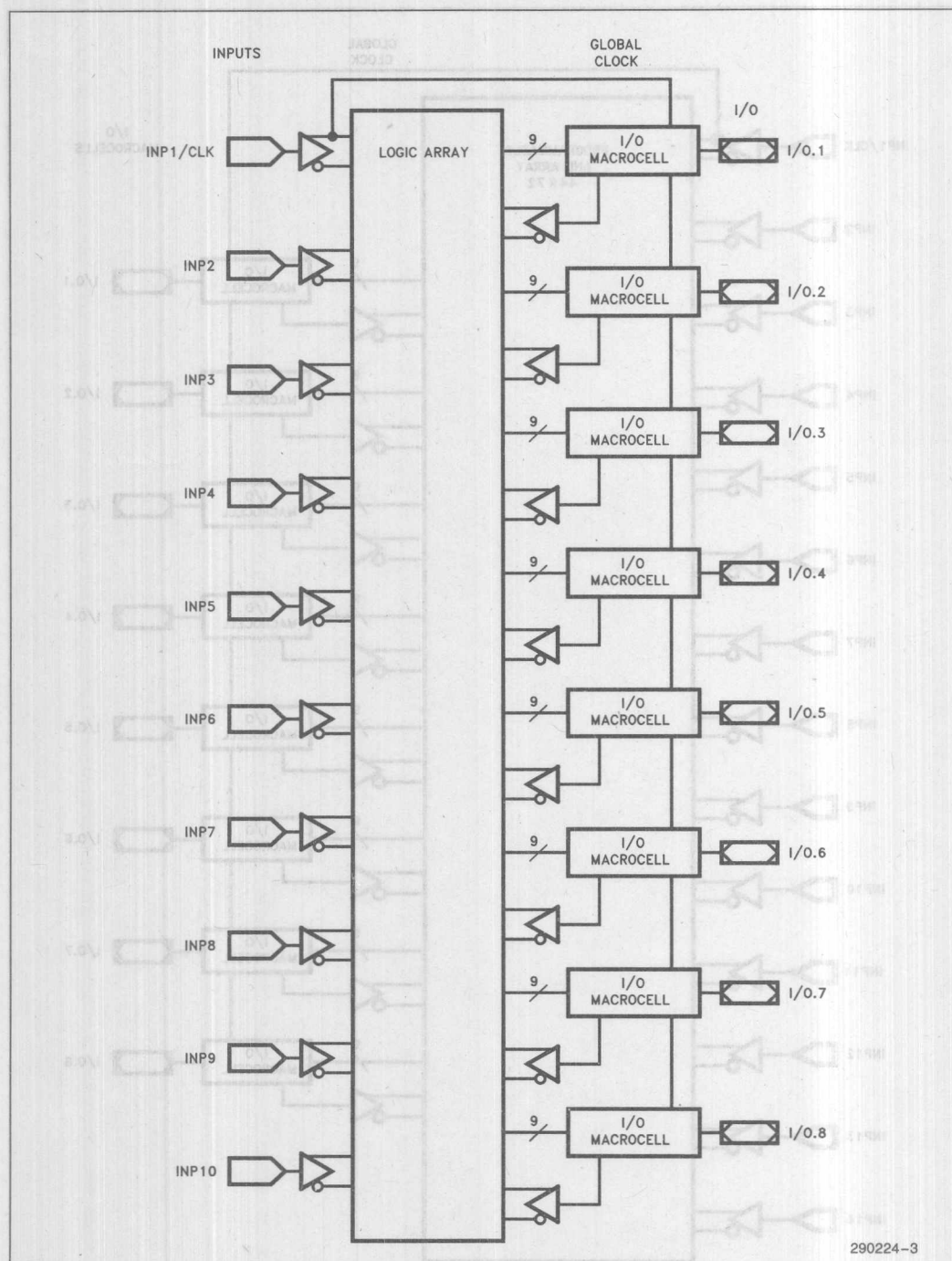


Figure 2. 85C220 Global Architecture



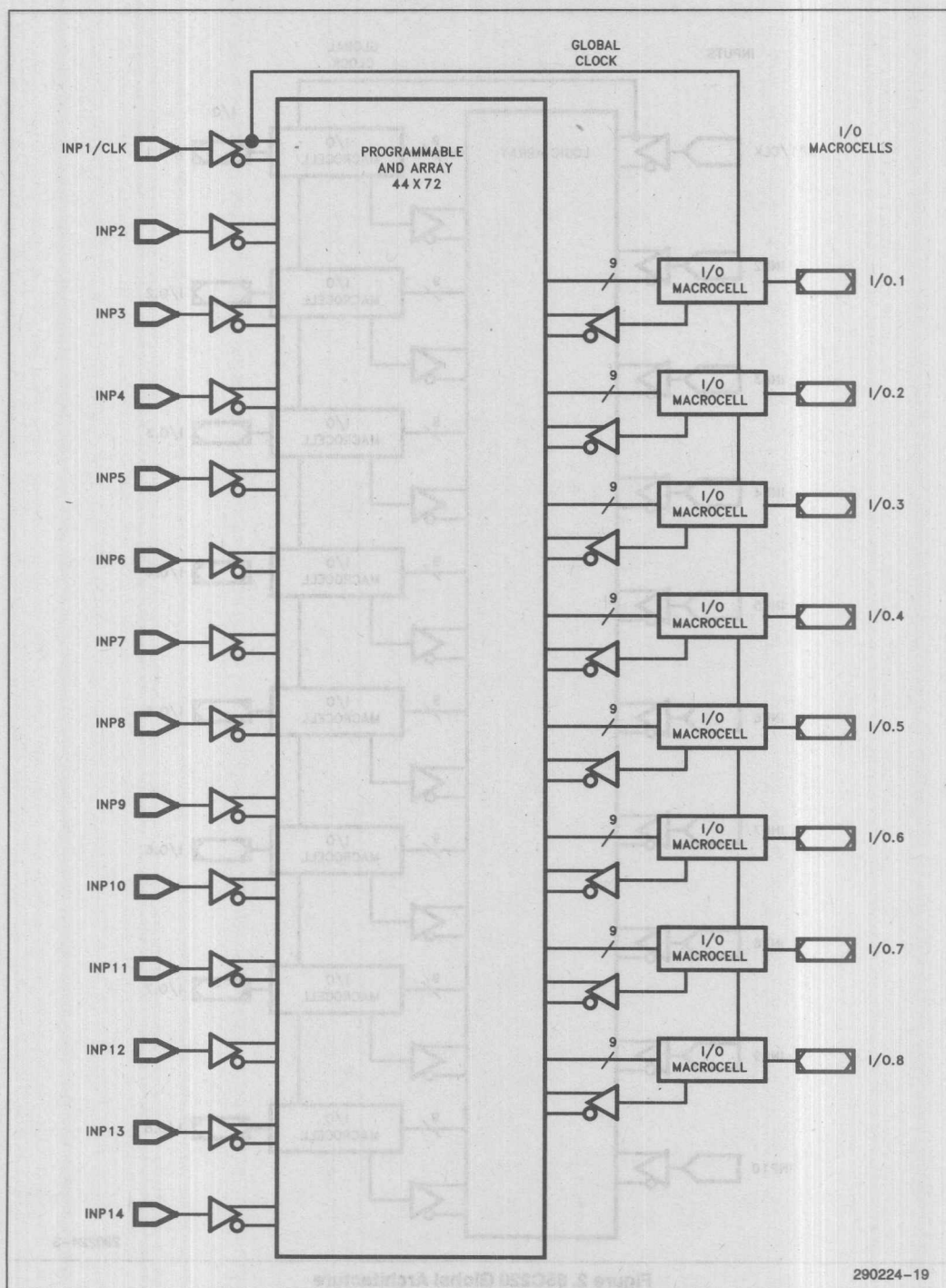
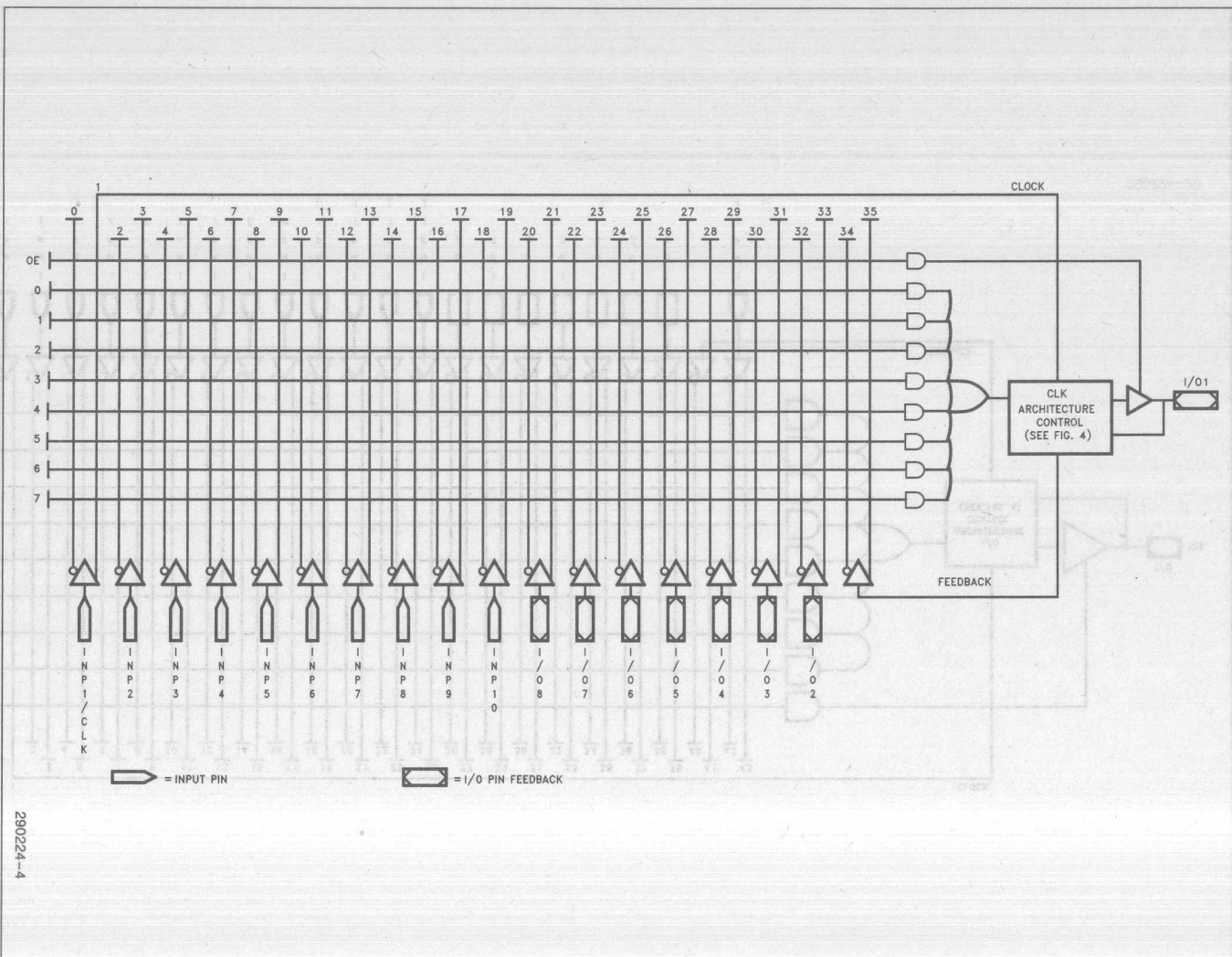


Figure 3. 85C224 Global Architecture



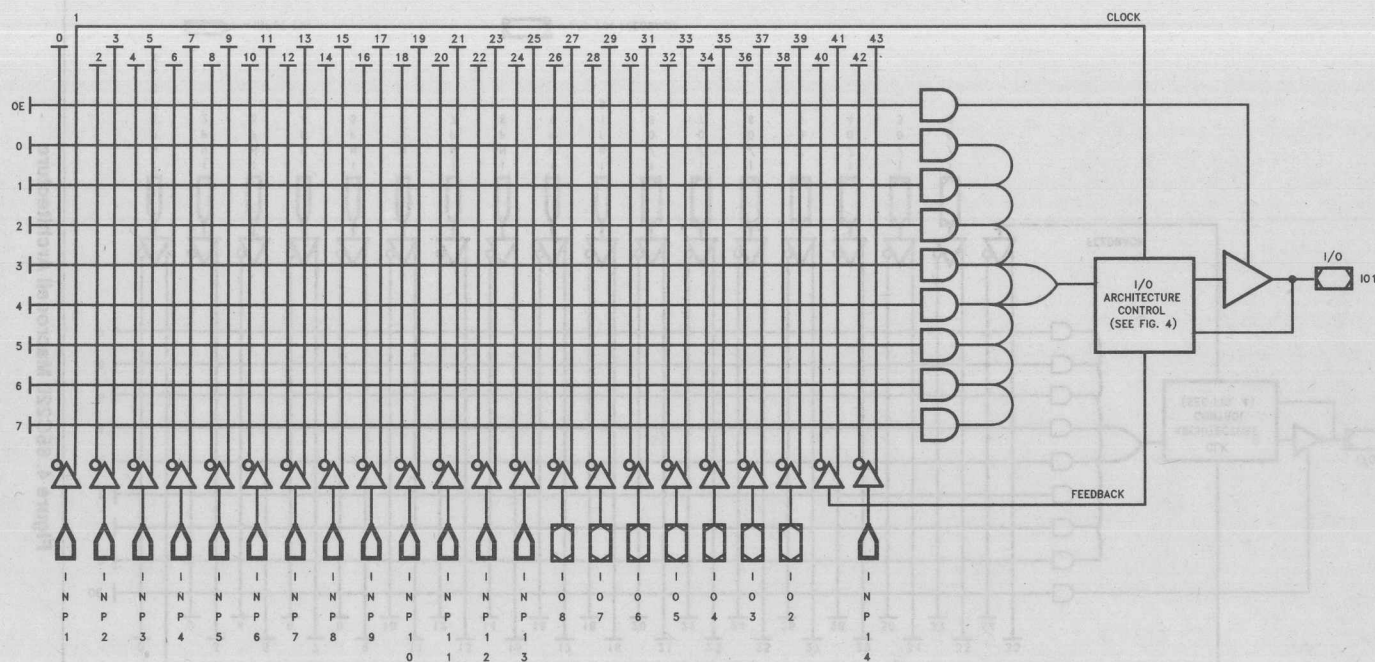


Figure 5. 85C224 Macrocell Architecture

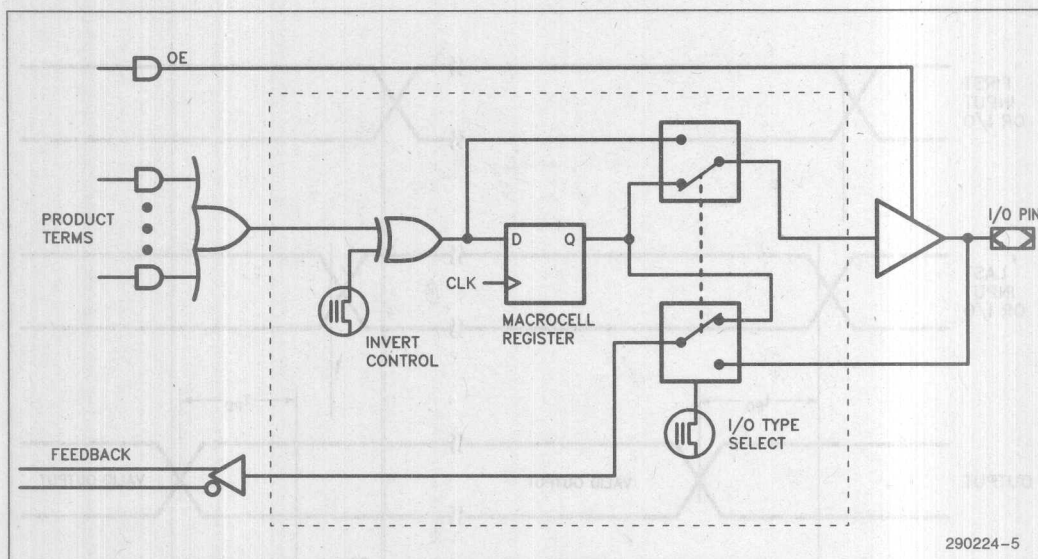


Figure 6. 85C220/85C224 I/O Control Architecture

## AUTOMATIC STAND-BY MODE

The 85C220/85C224 contains a programmable bit, the Turbo Bit, that optimizes operation either for speed or for power savings. When the Turbo Bit is programmed (TURBO = ON), the device is optimized for maximum speed. When the Turbo Bit is not programmed (TURBO = OFF), the device is optimized for power savings by entering standby mode during periods of inactivity.

Figure 7 shows the device entering standby mode approximately 75 ns after the last input or I/O transition. When the next input or I/O transition is detected, the device returns to active mode. Wakeup time adds an additional 20 ns to the propagation delay through the device as measured from the first transition. No delay will occur if an output is dependent on more than one input and the last of the inputs changes after the device has returned to active mode.

After erasure, the Turbo Bit is unprogrammed (OFF); automatic standby mode is enabled. When the Turbo Bit is programmed (ON), the device never enters standby mode.

## POWER-ON CHARACTERISTICS

85C220/85C224 inputs and outputs begin responding 1  $\mu$ s (max.) after  $V_{CC}$  power-up ( $V_{CC} = 4.75V$ ) or after a power-loss/power-up sequence. All macrocells programmed as registers will be set to a logic low.

## ERASED STATE CHARACTERISTICS

Prior to programming or after erasure, the I/O structure is configured for combinatorial active low output with input (pin) feedback.

Erase time for the 85C220/85C224 is 1 hour at 12,000  $\mu$ Wsec/cm<sup>2</sup> with a 2537Å UV lamp.

Erase characteristics of the device are such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000Å. It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000Å–4000Å range. Data shows that constant exposure to room level fluorescent lighting could erase the typi-



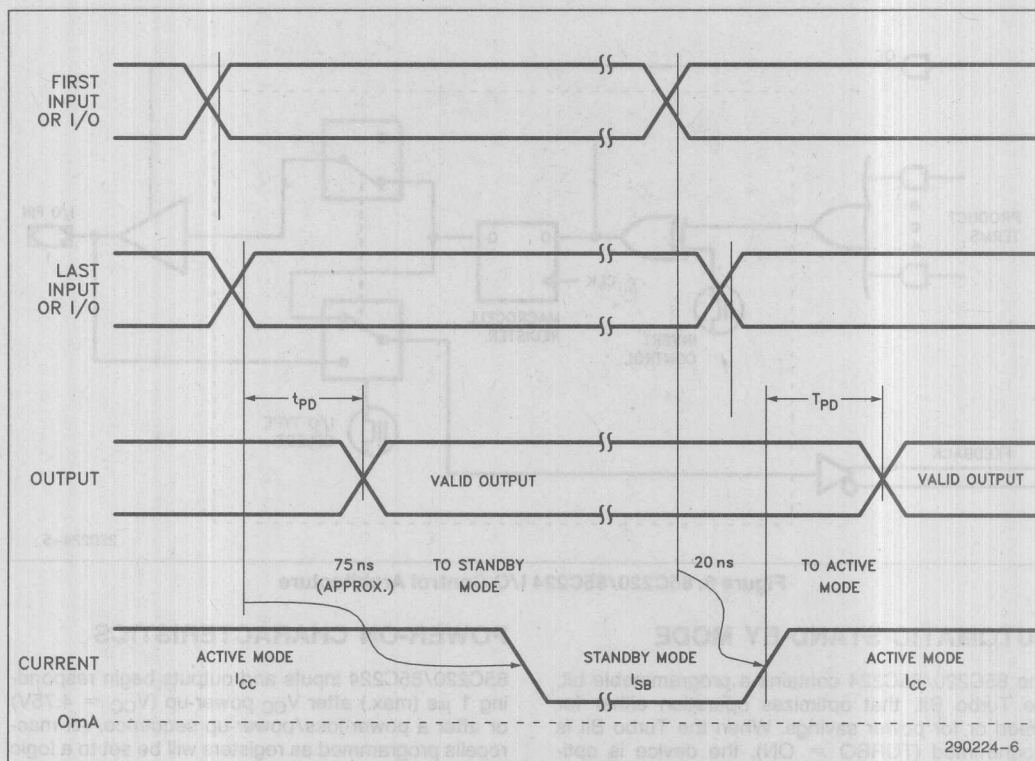


Figure 7. 85C220/85C224 Standby and Active Mode Transitions

cal 85C220/85C224 in approximately six years, while it would take approximately two weeks to erase the device when exposed to direct sunlight. If the device is to be exposed to these lighting conditions for extended periods of time, conductive opaque labels should be placed over the device window to prevent unintentional erasure.

The recommended erasure procedure for the 85C220/85C224 is exposure to shortwave ultraviolet light with a wavelength 2537Å. The integrated dose (i.e., UV intensity x exposure time) for erasure should be a minimum of forty (40) Wsec/cm<sup>2</sup>. The erasure time with this dosage is approximately 1 hour using an ultraviolet lamp with a 12,000 μW/cm<sup>2</sup> power rating. The device should be placed within 1 inch of the lamp tubes during exposure. The maximum integrated dose the 85C220/85C224 can be exposed to without damage is 7258 Wsec/cm<sup>2</sup> (1 week at 12,000 μW/cm<sup>2</sup>). Exposure to high intensity UV light for longer periods may cause permanent damage to the device.

### Intelligent Programming Algorithm

The 85C220/85C224 support the Intelligent Programming Algorithm, which rapidly programs Intel EPDs, and many of Intel's microcontrollers and EPROMs while maintaining a high degree of reliability. It is particularly suited for production programming environments. This method decreases the overall programming time while reliability is ensured as the incremental programming margin of each bit has been verified during programming. Programming voltage and waveform specifications are available by request from Intel to support device programming.

### LATCH-UP IMMUNITY

All of the input, output, and clock pins of the device have been designed to resist latch-up which is inherent in inferior CMOS structures. The 85C220/85C224 is designed with Intel's proprietary 1-micron

CHMOS IIIIE EPROM process. Thus, each of the pins will not experience latch-up with currents up to  $\pm 100$  mA and voltages ranging from  $-0.5$  V to ( $V_{CC} + 0.5$  V). The programming pin is designed to resist latch-up to the 13.5V maximum device limit.

## DESIGN RECOMMENDATIONS

For proper operation, it is recommended that all input and output pins be constrained to the voltage range  $GND \leq (V_{IN} \text{ or } V_{OUT}) \leq V_{CC}$ . All unused inputs and I/Os should be tied high or low to minimize power consumption (do not leave them floating). On the 85C224 PLCC package, the optional power pin (pin 1) and optional ground pins (8, 15, and 22) may be connected to power and ground, respectively, to reduce output switching noise. A high-speed power supply decoupling capacitor of at least  $0.2 \mu F$  must be connected directly between the  $V_{CC}$  and GND pins.

As with all CMOS devices, ESD handling procedures should be used with the devices to prevent damage to the devices during programming, assembly, and test.

## COMPILER SUPPORT

The 85C220 and 85C224 are supported by Intel's PLDshell Plus software as well as third-party logic compilers such as ABEL\*, CUPL\*, PLDDesigner\*, Log/IC\*, etc.

PLDshell Plus software is a free design package that accepts PALASM® 2-compatible source files. PLDshell Plus software allows you to design in a familiar language and to functionally simulate your design. You can also invoke third-party design packages directly from the PLDshell Run menu.

iPLS II includes the LOC (Logic Optimizing Compiler) and APT (Advanced Programming Tool). For detailed information on iPLS II, refer to the iPLDS II Data Sheet, order number: 290134.

The following ADF primitives are supported by this device:

INP	RONF
CONF	RORF
COIF	NORF

## PROGRAMMING SUPPORT

Programming for the 85C220 is supported by APT on the GUPI 20D20J Programming Adaptor using either an iUP-PC Personal Programmer or an iUP-200A/201A Universal Programmer. Programming for the 85C224 is supported by the same software/platforms using the GUPI 24D28J Programming Adaptor.

85C220/85C224 programming support is also provided by third-party programmer companies such as Data I/O, Logical Devices, STAG, etc. Please refer to the "Programming Support" lists in the *Programmable Logic* handbook for complete information and vendor contacts.

## ORDERING INFORMATION

f <sub>CNT1</sub> (MHz)	f <sub>MAX</sub> (MHz)	t <sub>PD</sub> (ns)	Order Code 20-Pin	Order Code 24-Pin	Package	Operating Range
100	115	7.5	N85C220-100	N85C224-100	PLCC	Commercial
80	111	10	D85C220-80	D85C224-80	*CerDIP	Commercial
			P85C220-80	P85C224-80	PDIP	
			N85C220-80	N85C224-80	PLCC	
66	90.9	12	D85C220-66	D85C224-66	*CerDIP	Commercial
			P85C220-66	P85C224-66	PDIP	
			N85C220-66	N85C224-66	PLCC	

\*Windowed CerDIP package allows UV erase.

\*ABEL is a trademark of Data I/O, Corporation. CUPL is a trademark of Logical Devices, Inc. PLDesigner is a trademark of MINC, Inc.

**ABSOLUTE MAXIMUM RATINGS\***

Supply Voltage ( $V_{CC}$ )<sup>(1)</sup> ..... -2.0V to +7.0V  
 Programming Supply  
 Voltage ( $V_{PP}$ )<sup>(1)</sup> ..... -2.0V to +13.5V  
 D.C. Input Voltage ( $V_I$ )<sup>(1,2)</sup> ... -0.5V to  $V_{CC} + 0.5V$   
 Storage Temperature ( $T_{STG}$ ) .... -65°C to +150°C  
 Ambient Temperature ( $T_{AMB}$ )<sup>(3)</sup> .. -10°C to +85°C

**NOTES:**

1. Voltages with respect to GND.
2. Minimum D.C. input is -0.5V. During transitions, the inputs may undershoot to -2.0V or overshoot to +7.0V for periods of less than 20 ns under no load conditions.
3. Under bias. Extended Temperature versions are also available.

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**\*WARNING:** *Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

**RECOMMENDED OPERATING CONDITIONS**

Symbol	Parameter	Min	Max	Units
$V_{CC}$	Supply Voltage	4.75	5.25	V
$V_{IN}$	Input Voltage	0	$V_{CC}$	V
$V_O$	Output Voltage	0	$V_{CC}$	V
$T_A$	Operating Temperature	0	+70	°C
$t_R$	Input Rise Time		500	ns
$t_F$	Input Fall Time		500	ns

**D.C. CHARACTERISTICS** ( $T_A = 0^\circ\text{C}$  to  $+70^\circ\text{C}$ ,  $V_{CC} = 5.0V \pm 5\%$ )**85C220/85C224-80 and -66**

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
$V_{IH}^{(4)}$	High Level Input Voltage	2.0		$V_{CC} + 0.3$	V	
$V_{IL}^{(4)}$	Low Level Input Voltage	-0.3		0.8	V	
$V_{OH}$	High Level Output Voltage	2.4			V	$I_O = -4.0$ mA D.C., $V_{CC} = \text{Min}$
$V_{OL}^{(5)}$	Low Level Output Voltage			0.45	V	$I_O = 12.0$ mA D.C., $V_{CC} = \text{Min}$
$I_I$	Input Leakage Current			$\pm 10$	$\mu\text{A}$	$V_{CC} = \text{Max}$ , $\text{GND} < V_{IN} < V_{CC}$
$I_{OZ}$	Output Leakage Current			$\pm 10$	$\mu\text{A}$	$V_{CC} = \text{Max}$ , $\text{GND} < V_{OUT} < V_{CC}$
$I_{SC}^{(6)}$	Output Short Circuit Current	-30		-120	mA	$V_{CC} = \text{Max}$ , $V_{OUT} = 0.5V$
$I_{SB}^{(7)}$	Standby Current		50	500	$\mu\text{A}$	$V_{CC} = \text{Max}$ , $V_{IN} = V_{CC}$ or GND, Standby Mode
$I_{CC}$	Power Supply Current (see $I_{CC}$ vs Frequency Graph)		2	5	mA	$V_{CC} = \text{Max}$ , $V_{IN} = V_{CC}$ or GND, No Load, $f_{IN} = 1$ MHz, Device Prog. as an 8-Bit Counter, Non-Turbo Mode
			35	50	mA	$f_{IN} = 15$ MHz, Active Mode
			45	60	mA	$f_{IN} = 80$ MHz, Active Mode

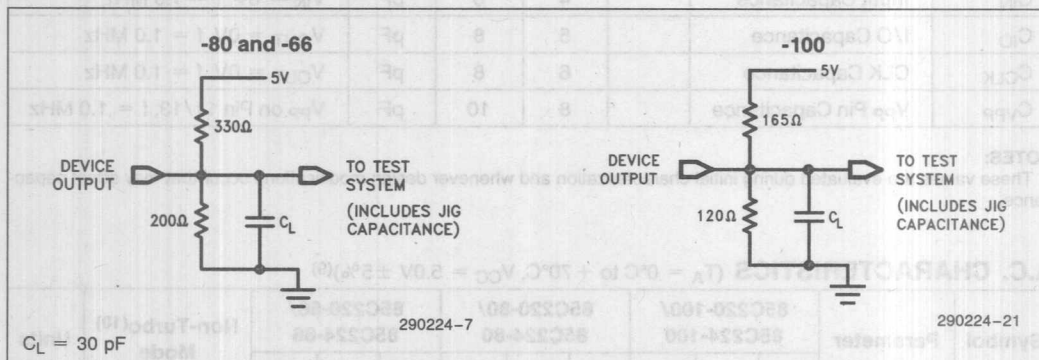
85C220/85C224-100

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
$V_{IH}^{(4)}$	High Level Input Voltage	2.0		$V_{CC} + 0.3$	V	
$V_{IL}^{(4)}$	Low Level Input Voltage	-0.3		0.8	V	
$V_{OH}$	High Level Output Voltage	2.4			V	I/O = -4.0 mA D.C., $V_{CC}$ = Min
$V_{OL}$	Low Level Output Voltage			0.45	V	I/O = 24.0 mA D.C., $V_{CC}$ = Min
$I_I$	Input Leakage Current			$\pm 10$	$\mu A$	$V_{CC}$ = Max, $GND < V_{IN} < V_{CC}$
$I_{OZ}$	Output Leakage Current			$\pm 10$	$\mu A$	$V_{CC}$ = Max, $GND < V_{OUT} < V_{CC}$
$I_{SC}^{(6)}$	Output Short Circuit Current	-30		-120	mA	$V_{CC}$ = Max, $V_{OUT} = 0.5V$
$I_{CC}$	Power Supply Current		60	90	mA	$V_{CC}$ = Max, $V_{IN} = V_{CC}$ or GND, No Load, $f_{IN} = 25$ MHz, Device Prog. as an 8-Bit Counter
			85	115	mA	$f_{IN} = 100$ MHz

NOTES:

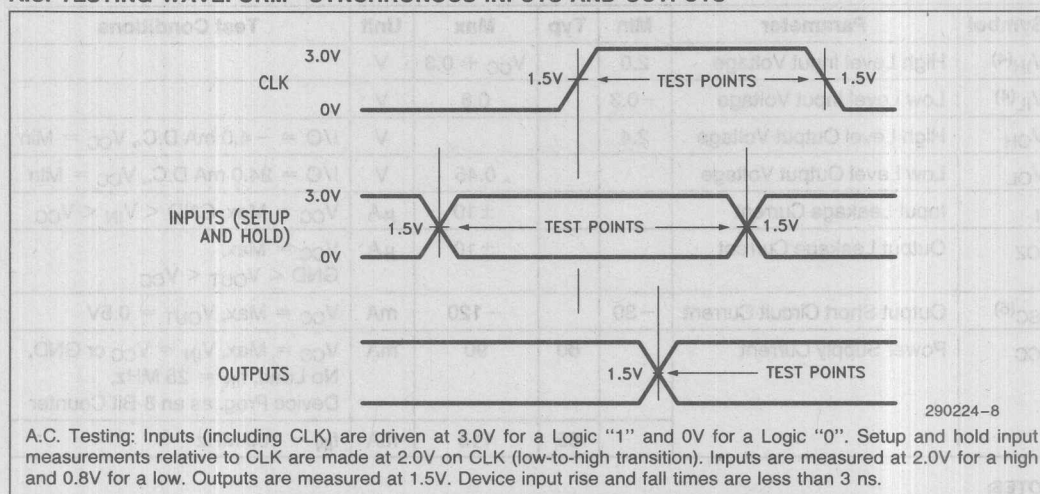
4. Absolute values with respect to device GND; all over and undershoots due to system or tester noise are included.
5. Maximum DC  $I_{OL}$  for the device (all 8 outputs) is 64 mA.
6. Not more than 1 output should be tested at a time. Duration of that test should not exceed 1 second.
7. In Non-Turbo Mode (TURBO=OFF), device enters standby mode approximately 75 ns after the last input transition.

A.C. TESTING LOAD CIRCUIT





# A.C. TESTING WAVEFORM—SYNCHRONOUS INPUTS AND OUTPUTS



## CAPACITANCE ( $T_A = 0^\circ\text{C}$ to $+70^\circ\text{C}$ ; $V_{CC} = 5.0\text{V} \pm 5\%$ )(8)

Symbol	Parameter	Min	Typ	Max	Unit	Conditions
$C_{IN}$	Input Capacitance		4	6	pF	$V_{IN} = 0\text{V}$ , $f = 1.0\text{ MHz}$
$C_{IO}$	I/O Capacitance		5	8	pF	$V_{OUT} = 0\text{V}$ , $f = 1.0\text{ MHz}$
$C_{CLK}$	CLK Capacitance		6	8	pF	$V_{OUT} = 0\text{V}$ , $f = 1.0\text{ MHz}$
$C_{VPP}$	$V_{PP}$ Pin Capacitance		8	10	pF	$V_{PP}$ on Pin 11/13, $f = 1.0\text{ MHz}$

### NOTES:

8. These values are evaluated during initial characterization and whenever design modifications occur that may affect capacitance.

## A.C. CHARACTERISTICS ( $T_A = 0^\circ\text{C}$ to $+70^\circ\text{C}$ , $V_{CC} = 5.0\text{V} \pm 5\%$ )(9)

Symbol	Parameter	85C220-100/ 85C224-100			85C220-80/ 85C224-80			85C220-66/ 85C224-66			Non-Turbo <sup>(10)</sup> Mode	Units
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
$t_{PD}^{(11)}$	Input or I/O to Output	3		7.5	4		10	4		12	+ 20	ns
$t_{pZX}^{(12)}$	Input or I/O to Output Enable	3		9	4		12	4		12	+ 20	ns
$t_{pXZ}^{(12)}$	Input or I/O to Output Disable	3		9	4		10	4		12	+ 20	ns

### NOTES:

9. Typical values are at  $T_A = +25^\circ\text{C}$ ,  $V_{CC} = 5\text{V}$ , Active Mode.

10. If device is operated in Standby Mode (Standby bit = Low) and the device is inactive for approximately 75 ns, increase time by amount shown for -80 and -66 only.

11. Measured with all eight outputs switching.

12.  $t_{pZX}$  and  $t_{pXZ}$  are measured at  $\pm 0.5\text{V}$  from steady state voltage as driven by specification output load.  $t_{pXZ}$  is measured with  $C_L = 5\text{ pF}$ . Measured with all eight outputs switching.

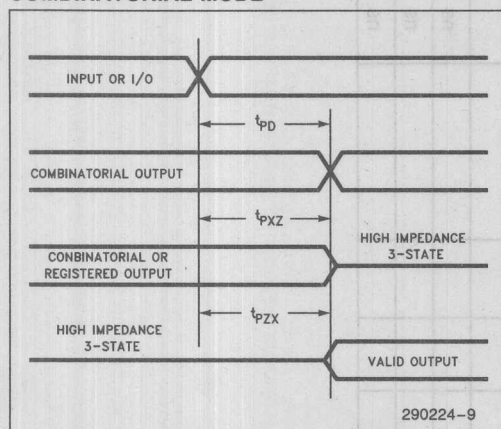
**SYNCHRONOUS CLOCK MODE** ( $T_A = 0^\circ\text{C to } 70^\circ\text{C}$ ,  $V_{CC} = 5V \pm 5\%$ )(9)

Symbol	Parameter	85C220-100/ 85C224-100			85C220-80/ 85C224-80			85C220-66 85C224-66			Non-Turbo <sup>(10)</sup> Mode	Units
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
$f_{CNT1}^{(11)}$	Maximum Counter Frequency 1/( $t_{SU} + t_{CO}$ )—External Feedback		111	100		100	80		80	66		MHz
$f_{CNT2}^{(11)}$	Maximum Counter Frequency 1/( $t_{CNT}$ )—Internal Feedback			115		111	100		90	83.3		MHz
$f_{MAX}^{(11)}$	Maximum Frequency (Pipelined) 1/( $t_{CW}$ )—No Feedback			115		125	111		100	90.9		MHz
$t_{SU}$	Input or I/O Setup Time to CLK	4.5			7			9			+ 20	ns
$t_H$	Input or I/O Hold Time from CLK	0			0			0				ns
$t_{CO1}$	CLK High to Output Valid	3		5.5	1.5(13)		5.5(11)	1.5(13)		6(11)		ns
$t_{CO2}$	CLK High to Output Valid Fed through Comb. Macrocell	4.5		10	4.5		13	4.5		15	+ 20	ns
$t_{CNT}^{(11)}$	Macrocell Output Feedback to Macrocell Input—Internal Path	10			10			12			+ 20	ns
$t_{CL}$	CLK Low Time	4			4			5				ns
$t_{CH}$	CLK High Time	4			4			5				ns
$t_{CW}$	CLK Period	10			9			11				ns

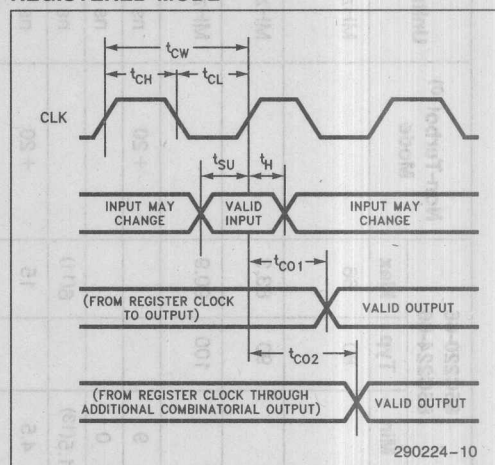
**NOTE:**

13.  $t_{CO1}$  min. is measured with one output switching,  $T_A = 0^\circ\text{C}$ ,  $V_{CC} = 5.25V$ .

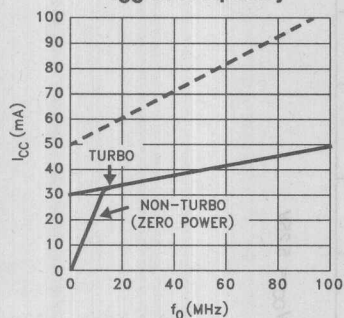
# COMBINATORIAL MODE



# REGISTERED MODE

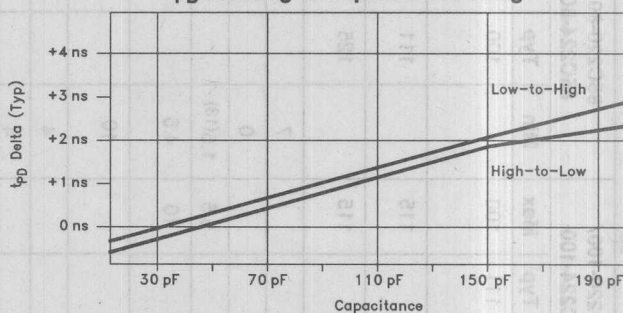


85C220/85C224  
 $I_{CC}$  vs Frequency



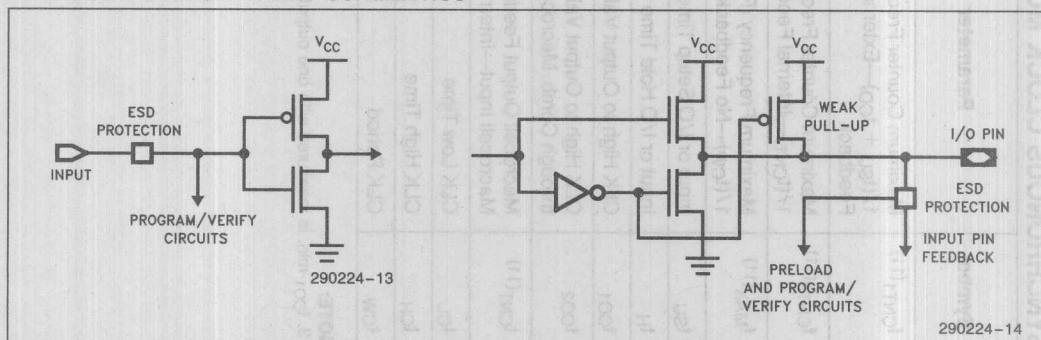
Conditions:  
 $T_A = 25^\circ\text{C}$   
 $V_{CC} = 5.25\text{V}$

85C220/85C224  
 $t_{PD}$  Derating vs Capacitive Loading

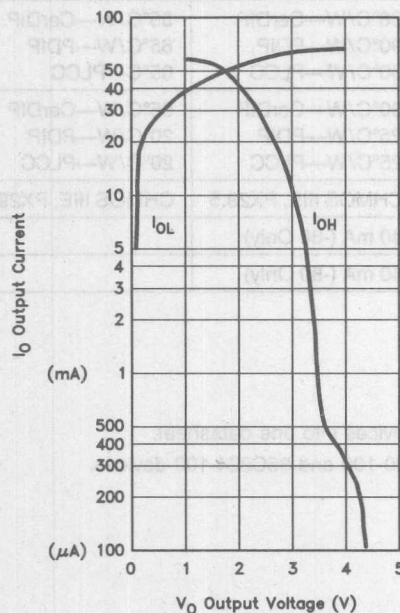


Conditions:  
 $T_A = 70^\circ\text{C}$   
 $V_{CC} = 4.75\text{V}$

# INPUT/OUTPUT EQUIVALENT SCHEMATICS



85C220/85C224 Output Drive Current  
in Relation to Voltage



290224-15

Conditions:  
 $T_A = +80^\circ C$   
 $V_{CC} = 4.75V$

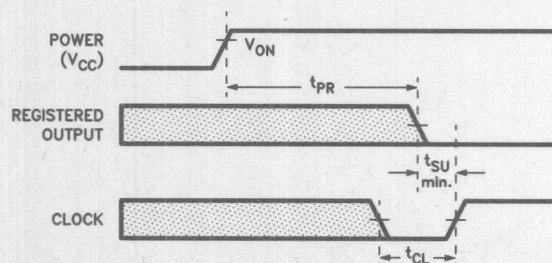
## Power-Up Reset

Internal power-up reset circuits ensure that all flip-flops will be reset to a logic 0 after the device has powered up. Because  $V_{CC}$  rise can vary significantly from one application to another,  $V_{CC}$  rise must be monotonic.

## POWER-UP RESET CHARACTERISTICS

Symbol	Parameter	Value
$t_{PR}$	Power-Up Reset Time	1000 ns Max.
$V_{ON}$	Turn-On Voltage	4.75V

## POWER-UP RESET



290224-16



## PACKAGE/TECHNOLOGY SPECIFICATIONS

Description	85C220	85C224
$\Theta_{Ja}$ —Junction-to-Ambient Thermal Resistance	68°C/W—CerDIP 90°C/W—PDIP 90°C/W—PLCC	55°C/W—CerDIP 65°C/W—PDIP 65°C—PLCC
$\Theta_{Jc}$ —Junction-to-Case Thermal Resistance	30°C/W—CerDIP 25°C/W—PDIP 25°C/W—PLCC	55°C/W—CerDIP 20°C/W—PDIP 20°C/W—PLCC
Process	CHMOS IIIIE, PX29.5	CHMOS IIIIE, PX29.5
$I_{CC}$ Hot—Ambient @70°C	40 mA (-80 Only)	
$I_{CC}$ Typical—Ambient @25°C	40 mA (-80 Only)	

## REVISION HISTORY

-003 to -004

Combined 85C220 and 85C224 devices into one datasheet.

Addition of specification for 85C220-100 and 85C224-100 devices.

# 85C220/85C224-7 AND -10

## COMBINATORIAL OPTIMIZED TIMING

### FAST 1-MICRON CHMOS

### 8-MACROCELL $\mu$ PLDs

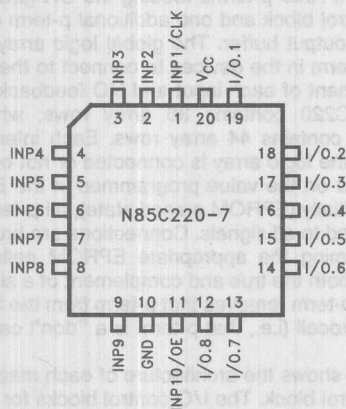
These Combinatorial Optimized Timing  $\mu$ PLDs Offer Superior Design Features:

- High-Performance Low-Power Upgrade for SSI/MSI Logic and Bipolar/CMOS PLDs in Intel386™, i486™, i860™, 80960 Series and Other High-Performance Systems
- Performance/Power Upgrade for -7 PALs\*/GALs\*; Superset of Common 20-Pin and 24-Pin PAL/GAL Architectures
- $t_{PD}$  7.5 ns, 74 MHz Frequency with External Feedback, 100 MHz Frequency with Internal Feedback and with No Feedback
- Up to 18/22 Inputs and 8 Outputs (18/22 Inputs = 10/14 Dedicated and 8 I/O)
- 8 I/O Macrocells with Programmable I/O Architecture (Register/Combinatorial)

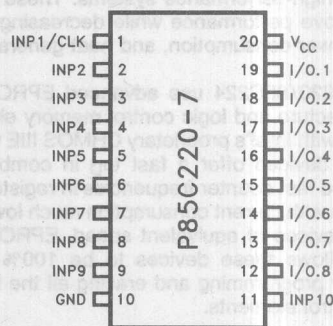
- 8 P-Terms, Selectable SOP Invert, OE P-Term for Each Macrocell
- "Half-Power"  $I_{CC} = 90$  mA Max. at 25 MHz, 105 mA Max. at 74 MHz
- Output Buffers Optimized for Low-Noise Operation
- Extensive Software and Programming Support via Intel and Third-Party Tools
- 1-Micron CHMOS\* IIIE EPROM Technology in OTP Package; 100% Generically Tested Logic Array
- Programmable "Security Bit" Allows Total Protection of Proprietary Designs
- Available in 20-Pin/28-Pin PLCC and 20-Pin/24-Pin PDIP Packages (PDIP220 Available Q2'93, PDIP224 Available Q1'93)

(See Packaging Spec., Order Number 240800, Package Types N and P)

2



290417-1



290417-14

Figure 1a. Pinout Diagrams

\*GAL is a registered trademark of Lattice Semiconductor Corporation.

\*PAL is a registered trademark of Advanced Micro Devices.

\*CHMOS is a patented process of Intel Corporation.

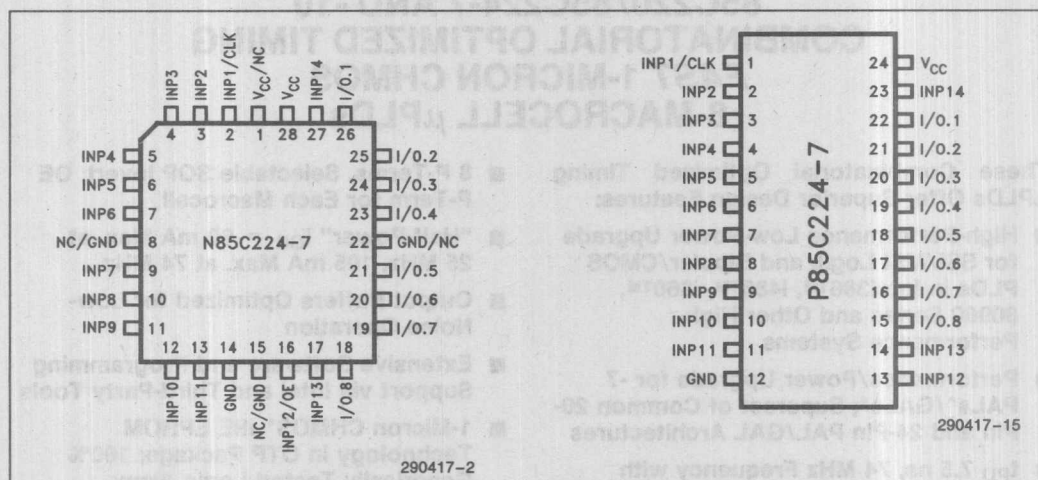


Figure 1b. Pinout Diagrams

## INTRODUCTION

The Intel 85C220-7 and 85C224-7 1-micron CHMOS  $\mu$ PLDs (Microcomputer Programmable Logic Devices) are superset devices capable of upgrading 7.5 ns PALs/GALs and 74-series LS and CMOS SSI and MSI logic devices in high-performance microcomputer systems. The inherent speed of the devices together with their lower power demands and plastic packaging make the 85C220 and 85C224 ideal production vehicles for high-volume manufacturing of high-performance systems. These devices can improve performance while decreasing system noise, power consumption, and heat generation.

The 85C220/85C224 use advanced EPROM cells as architecture and logic control memory elements. Coupled with Intel's proprietary CHMOS IIIE technology, the devices offer a fast  $t_{PD}$  in combinatorial mode, and fast counter frequencies in registered applications with current consumption much lower than bipolar devices of equivalent speed. EPROM technology allows these devices to be 100% factory tested by programming and erasing all the EPROM logic control elements.

## ARCHITECTURE DESCRIPTION

The architecture of the devices is based on the SOP (Sum of Products) PAL structure with a programmable AND array feeding into a fixed OR array. Programmable macrocells allow the device to accommodate both combinatorial and sequential logic functions. Each macrocell is individually programmable for combinatorial or registered output. An invert option on the SOP allows each output to be configured as an active-high or active-low output.

As shown in Figure 2, the 85C220 contains 10 dedicated inputs and 8 I/O pins. Figure 3 shows the 85C224, which contains 14 dedicated inputs and 8 I/O pins. On both devices, each I/O pin can be individually programmed to function as an input, output, or bidirectional I/O pin. Associated with each I/O pin is a programmable macrocell.

Figure 4 shows the structure of the 85C220 macrocell; Figure 5 shows the 85C224 macrocell. Each macrocell includes a p-term (product term) block with eight AND p-terms feeding the OR gate of the I/O control block and one additional p-term controlling the output buffer. The global logic array allows each p-term in the devices to connect to the true or complement of each input and I/O feedback signal. The 85C220 contains 36 array rows, while the 85C224 contains 44 array rows. Each intersecting point in the logic array is connected or not connected based on the value programmed in the EPROM array. Initially (EPROM erased state), all p-terms are connected to all signals. Connections are broken by programming the appropriate EPROM cells. Connecting both the true and complement of a signal for a given p-term removes that p-term from the SOP for the macrocell (i.e., that p-term is a "don't care").

Figure 6 shows the architecture of each macrocell's I/O control block. The I/O control blocks for the devices are identical. The SOP input to the I/O control block can be inverted or non-inverted. The output can be registered or combinatorial. When registered output is selected, feedback to the logic array comes directly from the register (before the output buffer). When combinatorial output is selected, feedback comes from the I/O pin (after the output buffer) and can be used for bidirectional I/O. The register is a D-type register that clocks on the rising edge of the global CLK.

**85C220/85C224-PLD COMPATIBILITY**

The 85C224 is a logical superset of most high-speed 24-pin PAL/GAL devices. The I/O and logic sections of the device can be configured to emulate the devices listed. Designers can often replace multiple PALs with the 85C224 devices. In many cases 22V10 devices can also be replaced. The following list includes some of the devices with which 85C224 is compatible.

**Replacement/Upgrade****7 ns—20-Pin and 24-Pin**

Company	20-Pin Part	24-Pin Part
Intel	85C220-7	85C224-7
Lattice	GAL16V8B	GAL20V8B
AMD	PAL16L8	PAL20L8
AMD	PAL16R8	PAL20R8
AMD	PALCE16V8	PALCE20V8
National	PAL16L8	N/A
National	PAL16R8	N/A
Signetics	PLUS16L8	PLUS20L8
Signetics	PLUS16R8	PLUS20R8
TI	TIBPAL16L8	TIBPAL20L8
TI	TIBPAL16R8	TIBPAL20R8

**Replacement/Upgrade****10 ns—20-Pin and 24-Pin**

Company	20-Pin	24-Pin
Intel	85C220-10	85C224-10
Lattice	GAL16V8A	GAL20V8A
Lattice	GAL16V8B	GAL20V8B
AMD	PAL16L8	PAL20L8
AMD	PAL20L8	PAL20R8
AMD	PALCE16V8	PALCE20V8
National	PAL16L8	PAL20L8
National	PAL16R8	PAL20R8
National	GAL16V8A	GAL20V8A
Signetics	PLUS16L8	PLUS20L8
Signetics	PLUS16R8	PLUS20R8
TI	TIBPAL16L8	TIBPAL20L8
TI	TIBPAL16R8	TIBPAL20R8

**NOTES:**

- Easy Cross Programming for JEDEC compatibility.
- Extensive software and programming support by Intel and Third-Party Tools.



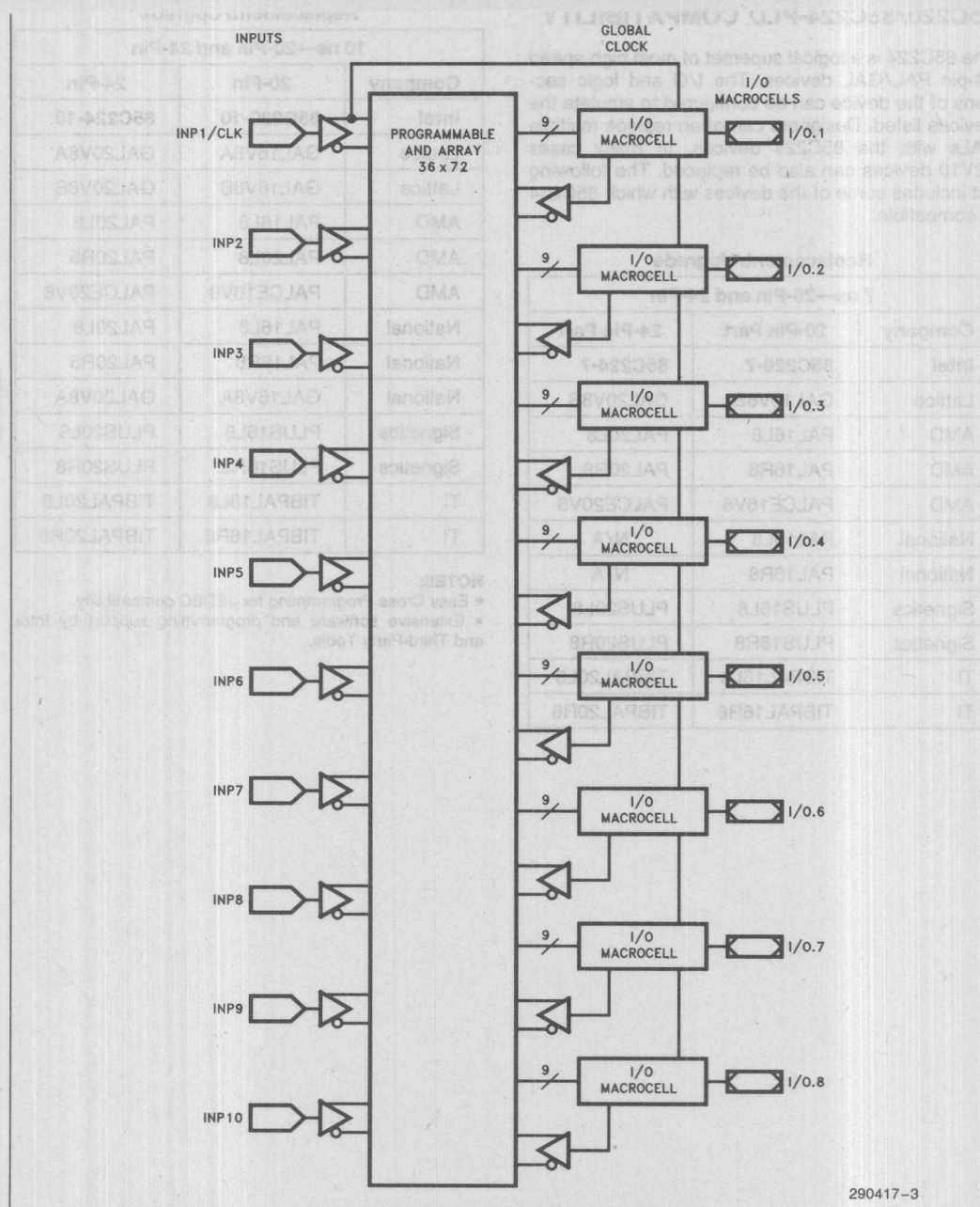


Figure 2. 85C220 Global Architecture

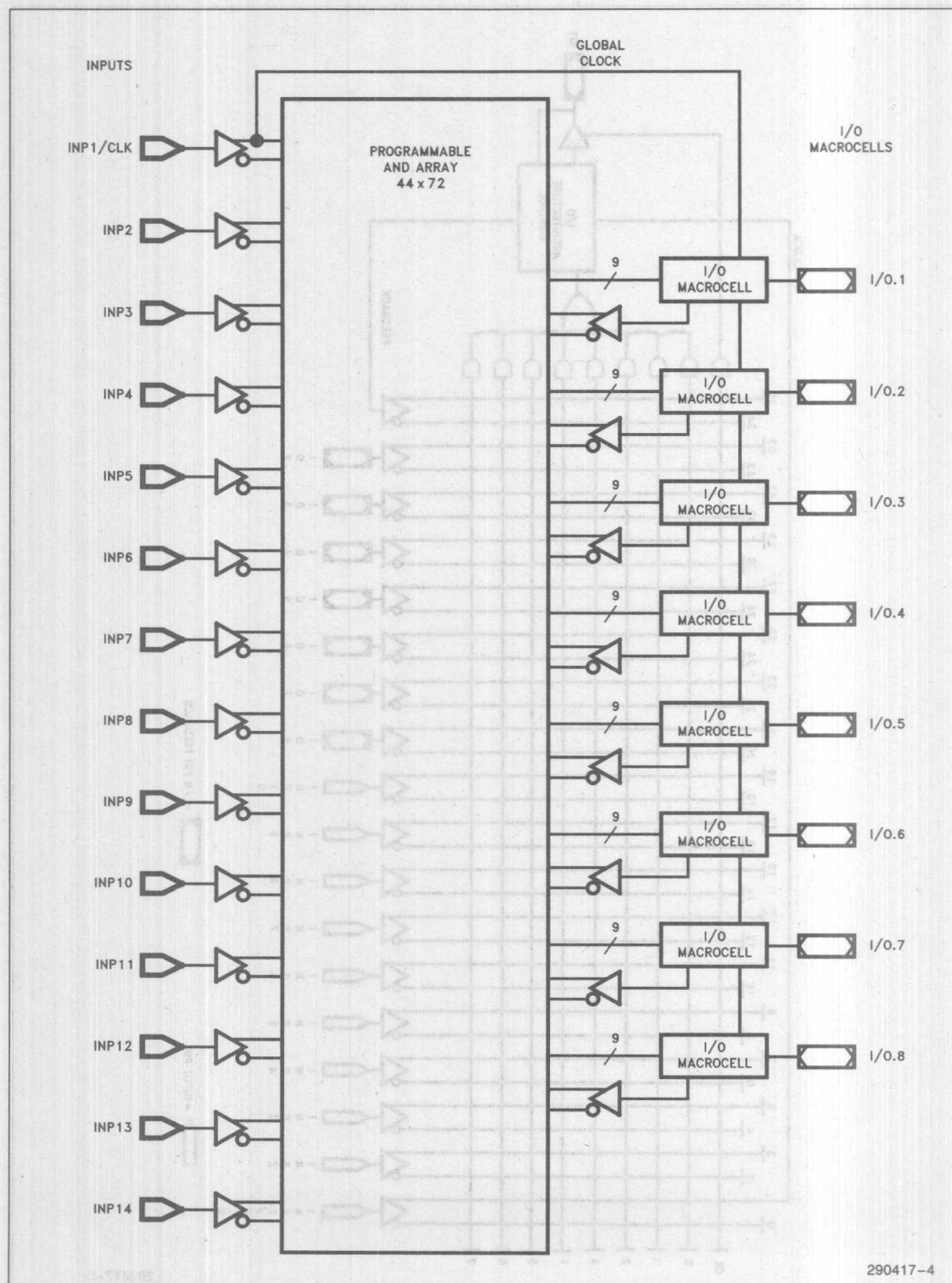


Figure 3. 85C224 Global Architecture

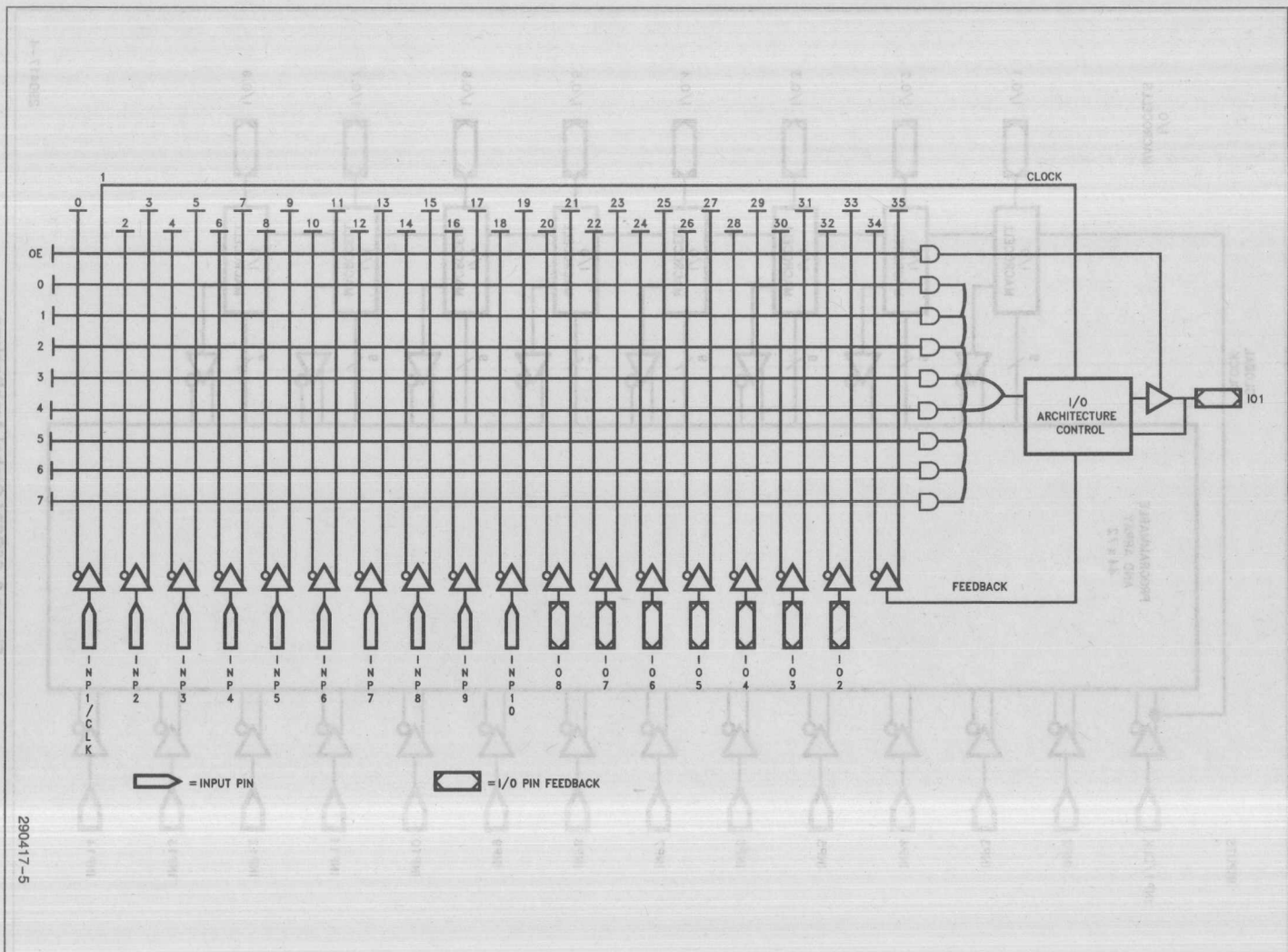
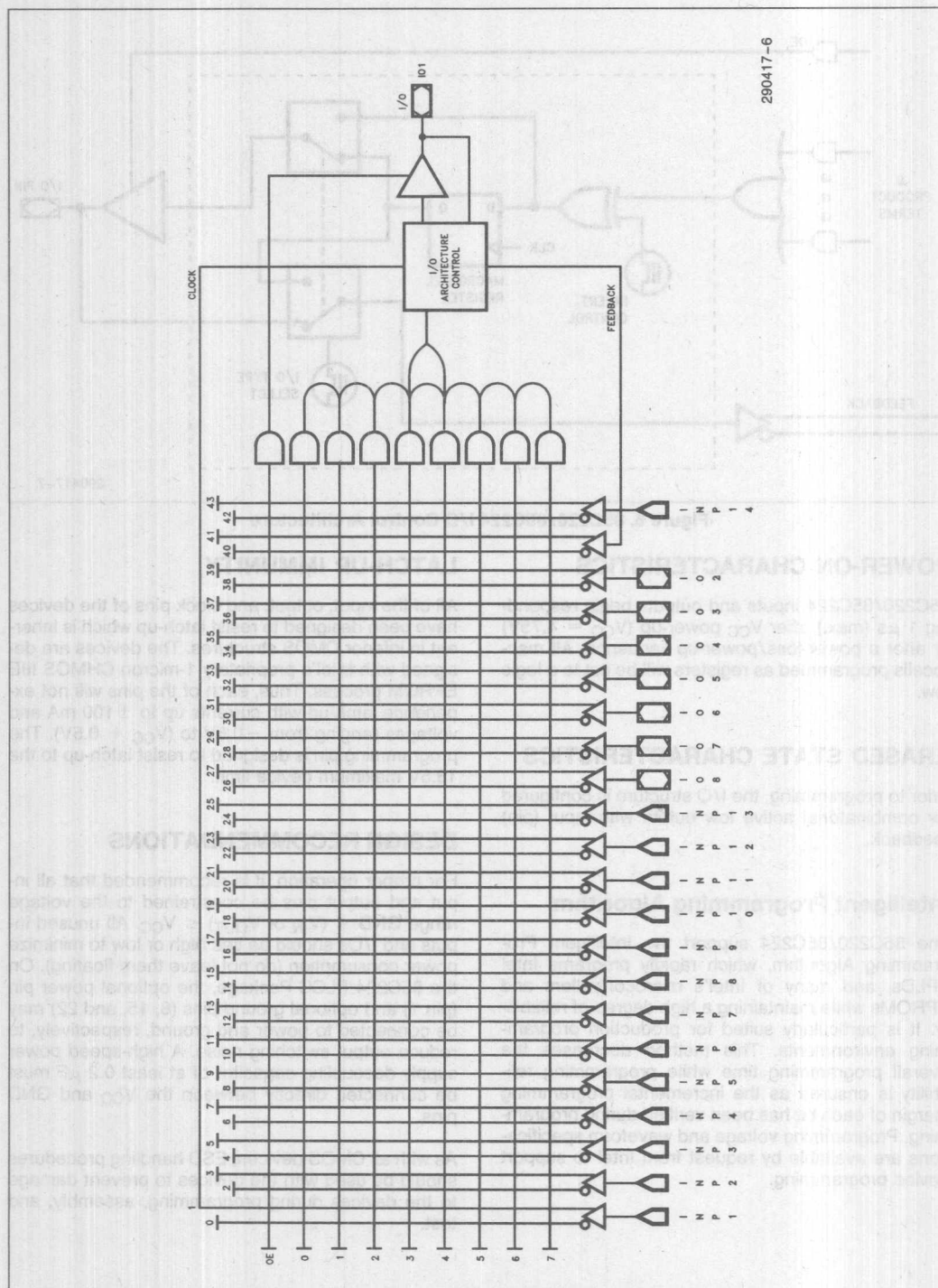


Figure 4. 85C220 Macrocell



290417-6

Figure 5. 85C224 Macrocell



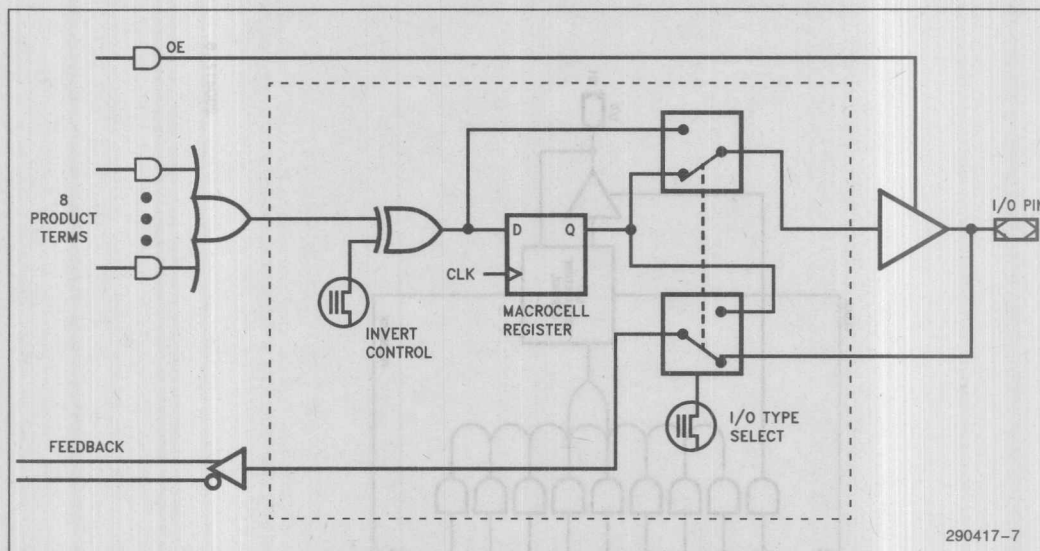


Figure 6. 85C220/85C224 I/O Control Architecture

## POWER-ON CHARACTERISTICS

85C220/85C224 inputs and outputs begin responding 1  $\mu$ s (max.) after  $V_{CC}$  power-up ( $V_{CC} = 4.75V$ ) or after a power-loss/power-up sequence. All macrocells programmed as registers will be set to a logic low.

## ERASED STATE CHARACTERISTICS

Prior to programming, the I/O structure is configured for combinatorial active low output with input (pin) feedback.

## Intelligent Programming Algorithm

The 85C220/85C224 support the Intelligent Programming Algorithm, which rapidly programs Intel EPLDs, and many of Intel's microcontrollers and EPROMs while maintaining a high degree of reliability. It is particularly suited for production programming environments. This method decreases the overall programming time while programming reliability is ensured as the incremental programming margin of each bit has been verified during programming. Programming voltage and waveform specifications are available by request from Intel to support device programming.

## LATCH-UP IMMUNITY

All of the input, output, and clock pins of the devices have been designed to resist latch-up which is inherent in inferior CMOS structures. The devices are designed with Intel's proprietary 1-micron CHMOS III E EPROM process. Thus, each of the pins will not experience latch-up with currents up to  $\pm 100$  mA and voltages ranging from  $-0.5V$  to  $(V_{CC} + 0.5V)$ . The programming pin is designed to resist latch-up to the 13.5V maximum device limit.

## DESIGN RECOMMENDATIONS

For proper operation, it is recommended that all input and output pins be constrained to the voltage range  $GND \leq (V_{IN} \text{ or } V_{OUT}) \leq V_{CC}$ . All unused inputs and I/Os should be tied high or low to minimize power consumption (do not leave them floating). On the 85C224 PLCC Package, the optional power pin (pin 1) and optional ground pins (8, 15, and 22) may be connected to power and ground, respectively, to reduce output switching noise. A high-speed power supply decoupling capacitor of at least 0.2  $\mu F$  must be connected directly between the  $V_{CC}$  and GND pins.

As with all CMOS devices, ESD handling procedures should be used with the devices to prevent damage to the devices during programming, assembly, and test.

## COMPILER SUPPORT

The 85C220 and 85C224 are supported by Intel's PLDshell Plus™ software and iPLS II (Intel Programmable Logic Software), as well as third-party logic compilers such as ABEL\*, CUPL\*, PLDDesigner\*, Log/IC\*, etc.

PLDshell Plus software is a free design package that accepts PALASM® 2-compatible source files. PLDshell Plus software allows you to design in a familiar language and to functionally simulate your design. You can also invoke third-party design packages directly from the PLDshell Run menu.

iPLS II includes the LOC (Logic Optimizing Compiler) and APT (Advanced Programming Tool). For detailed information on iPLS II, refer to the iPLDS II Data Sheet, order number: 290134.

The following ADF primitives are supported by this device:

INP	RONF
CONF	RORF
COIF	NORF

## PROGRAMMING SUPPORT

Programming for the 85C220 is supported by APT on the GUPI 20D20J Programming Adaptor using either an iUP-PC Personal Programmer or an iUP-200A/201A Universal Programmer. Programming for the 85C224 is supported by the same software/platforms using the GUPI 24D28J Programming Adaptor.

85C220/85C224 programming support is also provided by third-party programmer companies such as Data I/O, Logical Devices, STAG, etc. Please refer to the *Programmable Logic* handbook for complete information and vendor contacts.

2

## 85C220-7 ORDERING INFORMATION

f <sub>CNT1</sub>	f <sub>MAX</sub>	t <sub>PD1</sub> /t <sub>PD2</sub>	Order Code	Package	Operating Range
74	100	7.5/8.5	N85C220-7 P85C220-7	PLCC PDIP	Commercial
58.8	62.5	10	N85C220-10 P85C220-10	PLCC PDIP	Commercial

## 85C224-7 ORDERING INFORMATION

f <sub>CNT1</sub>	f <sub>MAX</sub>	t <sub>PD1</sub> /t <sub>PD2</sub>	Order Code	Package	Operating Range
74	100	7.5/8.5	N85C224-7 P85C224-7	PLCC PDIP	Commercial
58.8	62.5	10	N85C224-10 P85C224-10	PLCC PDIP	Commercial

ABEL™ is a trademark of Data I/O Corporation.

CUPL™ is a trademark of Logical Devices, Inc.

\*Intel386™, i486™, i860™ and PLDshell Plus™ are trademarks of Intel Corporation.

Log/IC™ is a trademark of ISDATA Corporation.

PLDDesigner™ is a trademark of MINC, Inc.

**ABSOLUTE MAXIMUM RATINGS\***

Supply Voltage ( $V_{CC}$ )<sup>(1)</sup> ..... -2.0V to +7.0V  
 Programming Supply  
 Voltage ( $V_{PP}$ )<sup>(1)</sup> ..... -2.0V to +13.5V  
 D.C. Input Voltage ( $V_I$ )<sup>(1,2)</sup> ... -0.5V to  $V_{CC} + 0.5V$   
 Storage Temperature ( $T_{stg}$ ) ..... -65°C to +150°C  
 Ambient Temperature ( $T_{amb}$ )<sup>(3)</sup> ... -10°C to +85°C

NOTICE: This data sheet contains preliminary information on new products in production. The specifications are subject to change without notice. Verify with your local Intel Sales office that you have the latest data sheet before finalizing a design.

*\*WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

**RECOMMENDED OPERATING CONDITIONS**

Symbol	Parameter	Min	Max	Units
$V_{CC}$	Supply Voltage	4.75	5.25	V
$V_{IN}$	Input Voltage	0	$V_{CC}$	V
$V_O$	Output Voltage	0	$V_{CC}$	V
$T_A$	Operating Temperature	0	+70	°C
$t_R$	Input Rise Time		500	ns
$t_F$	Input Fall Time		500	ns

**PACKAGE/TECHNOLOGY SPECIFICATIONS**

Description	Specification	
	85C220-7	85C220
①Ja—Junction-to-Ambient Thermal Resistance	90°C/W—PLCC, PDIP	65°C/W—PLCC, PDIP
①Jc—Junction-to-Case Thermal Resistance	25°C/W—PLCC, PDIP	20°C/W—PLCC, PDIP
Process	CHMOS III E, PX29.5	CHMOS III E, PX29.5

**D.C. CHARACTERISTICS** ( $T_A = 0^\circ\text{C}$  to  $+70^\circ\text{C}$ ,  $V_{CC} = 5.0V \pm 5\%$ )

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
$V_{IH}^{(4)}$	High Level Input Voltage	2.0		$V_{CC} + 0.3$	V	
$V_{IL}^{(4)}$	Low Level Input Voltage	-0.3		0.8	V	
$V_{OH}$	High Level Output Voltage	2.4			V	I/O = -4.0 mA D.C., $V_{CC} = \text{Min.}$
$V_{OL}$	Low Level Output Voltage			0.45	V	I/O = 24.0 mA D.C., $V_{CC} = \text{Min.}$
$I_I$	Input Leakage Current			$\pm 10$	$\mu\text{A}$	$V_{CC} = \text{Max.}$ , GND < $V_{IN}$ < $V_{CC}$
$I_{OZ}$	Output Leakage Current			$\pm 10$	$\mu\text{A}$	$V_{CC} = \text{Max.}$ , GND < $V_{OUT}$ < $V_{CC}$
$I_{SC}^{(5)}$	Output Short Circuit Current	-30		-120	mA	$V_{CC} = \text{Max.}$ , $V_{OUT} = 0.5V$

**D.C. CHARACTERISTICS** ( $T_A = 0^\circ\text{C}$  to  $+70^\circ\text{C}$ ,  $V_{CC} = 5.0\text{V} \pm 5\%$ ) (Continued)

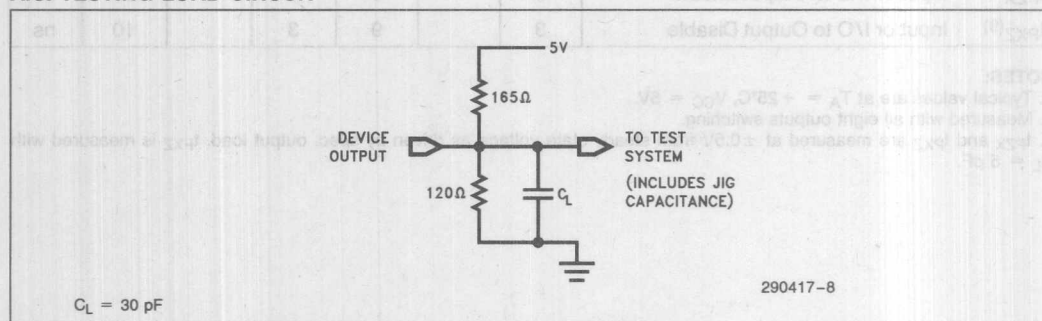
Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
$I_{CC}$	Power Supply Current (PLCC)		60	90	mA	$V_{CC} = \text{Max.}$ , $V_{IN} = V_{CC}$ or GND, No Load, $f_{IN} = 25\text{ MHz}$ , Device Prog. as 8-Bit Counter
			75	105	mA	$f_{IN} = 74\text{ MHz}$
	Power Supply Current (PDIP)		75	115	mA	$V_{CC} = \text{Max.}$ , $V_{IN} = V_{CC}$ or GND, No Load, $f_{IN} = 25\text{ MHz}$ , Device Prog. as 8-Bit Counter
			94	135	mA	$f_{IN} = 74\text{ MHz}$

**NOTES:**

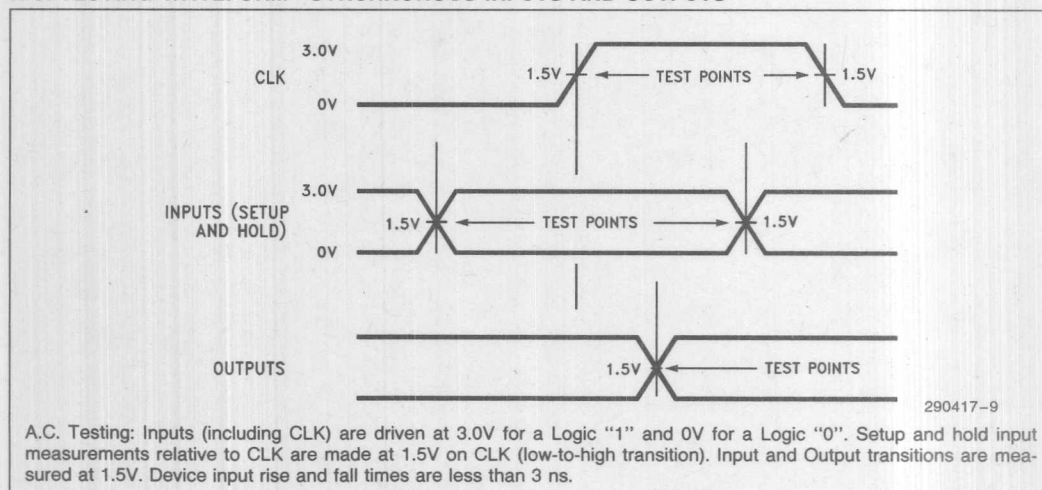
1. Voltages with respect to GND.
2. Minimum D.C. input is  $-0.5\text{V}$ . During transitions, the inputs may undershoot to  $-2.0\text{V}$  or overshoot to  $+7.0\text{V}$  for periods of less than 20 ns under no load conditions.
3. Under bias.
4. Absolute values with respect to device GND; all over and undershoots due to system or tester noise are included.
5. Not more than 1 output should be tested at a time. Duration of that test should not exceed 1 second.

2

**A.C. TESTING LOAD CIRCUIT**



**A.C. TESTING WAVEFORM—SYNCHRONOUS INPUTS AND OUTPUTS**





**CAPACITANCE** ( $T_A = 0^\circ\text{C}$  to  $+70^\circ\text{C}$ ;  $V_{CC} = 5.0\text{V} \pm 5\%$ )(6)

Symbol	Parameter	Min	Typ	Max	Unit	Conditions
$C_{IN}$	Input Capacitance		4	6	pF	$V_{IN} = 0\text{V}$ , $f = 1.0\text{ MHz}$
$C_{IO}$	I/O Capacitance		5	8	pF	$V_{OUT} = 0\text{V}$ , $f = 1.0\text{ MHz}$
$C_{CLK}$	CLK Capacitance		6	8	pF	$V_{OUT} = 0\text{V}$ , $f = 1.0\text{ MHz}$
$C_{VPP}$	$V_{PP}$ Pin Capacitance		8	10	pF	$V_{PP}$ on Pin 11/13, $f = 1.0\text{ MHz}$

**NOTE:**

6. These values are evaluated during initial characterization and whenever design modifications occur that may affect capacitance.

**A.C. CHARACTERISTICS** ( $T_A = 0^\circ\text{C}$  to  $+70^\circ\text{C}$ ,  $V_{CC} = 5.0\text{V} \pm 5\%$ )(7)

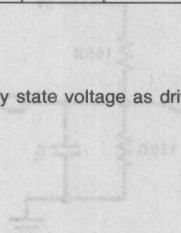
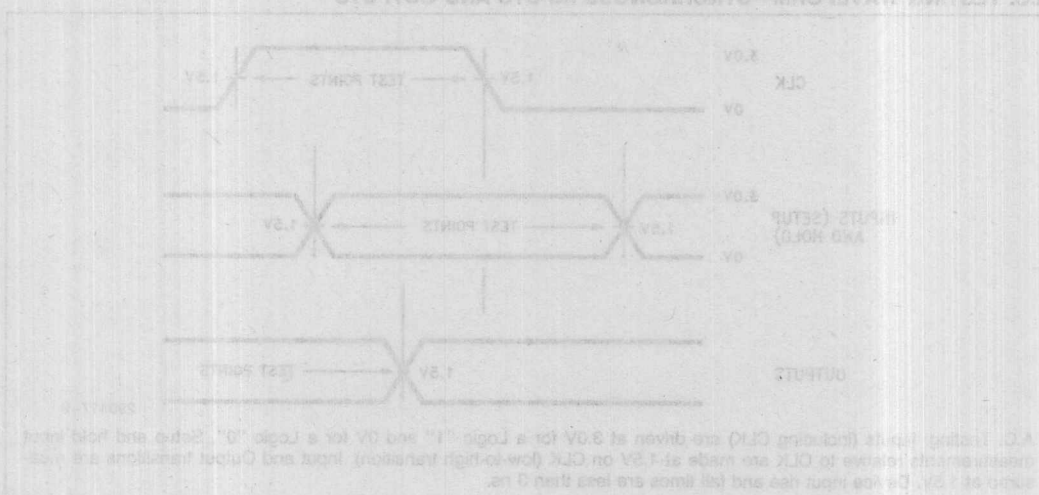
Symbol	Parameter	85C220-7/85C224-7			85C220-10/85C224-10			Units
		Min	Typ	Max	Min	Typ	Max	
$t_{PD1}^{(8)}$	Input or I/O to Output Valid, Invert On	3		7.5	3		10	ns
$t_{PD2}^{(8)}$	Input or I/O to Output Valid, Invert Off	3		8.5	3		10	ns
$t_{PZX}^{(9)}$	Input or I/O to Output Enable	3		9	3		10	ns
$t_{PXZ}^{(9)}$	Input or I/O to Output Disable	3		9	3		10	ns

**NOTES:**

7. Typical values are at  $T_A = +25^\circ\text{C}$ ,  $V_{CC} = 5\text{V}$ .

8. Measured with all eight outputs switching.

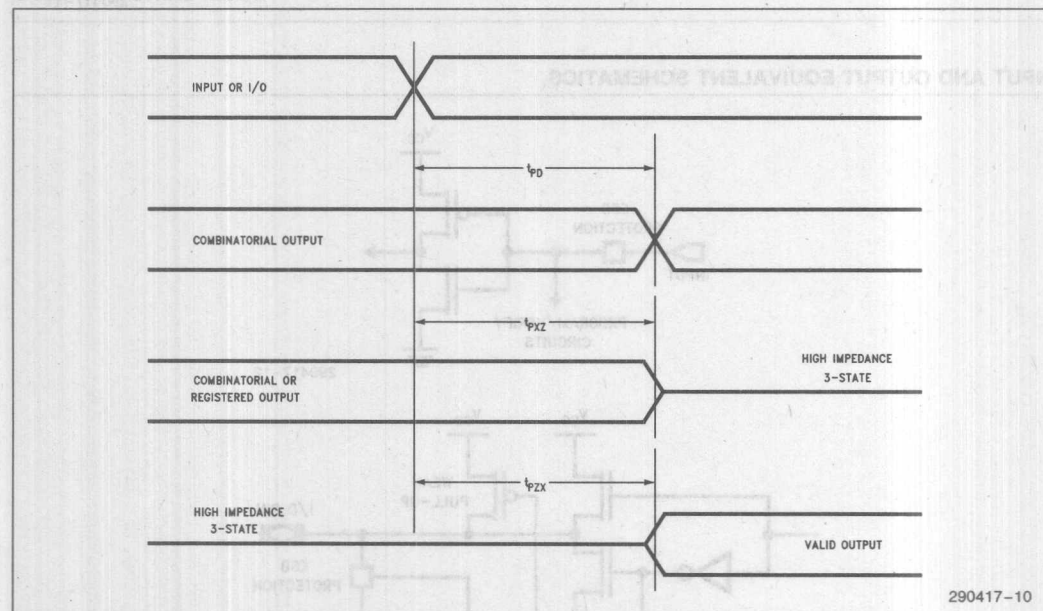
9.  $t_{PZX}$  and  $t_{PXZ}$  are measured at  $\pm 0.5\text{V}$  from steady state voltage as driven by spec. output load.  $t_{PXZ}$  is measured with  $C_L = 5\text{ pF}$ .

**A.C. TESTING WAVEFORM—SYNCHRONOUS INPUTS AND OUTPUTS**

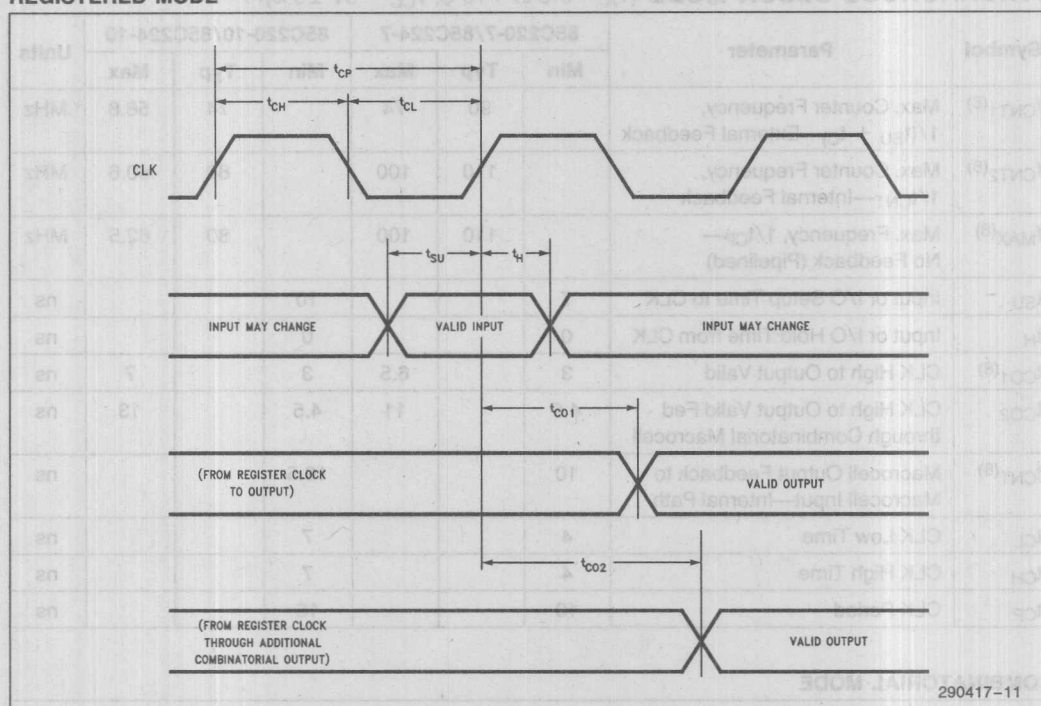
**SYNCHRONOUS CLOCK MODE** ( $T_A = 0^\circ\text{C}$  to  $+70^\circ\text{C}$ ,  $V_{CC} = 5V \pm 5\%$ )(7)

Symbol	Parameter	85C220-7/85C224-7			85C220-10/85C224-10			Units
		Min	Typ	Max	Min	Typ	Max	
$f_{CNT1}^{(8)}$	Max. Counter Frequency, $1/(t_{SU} + t_C)$ —External Feedback		90	74		74	58.8	MHz
$f_{CNT2}^{(8)}$	Max. Counter Frequency, $1/t_{CNT}$ —Internal Feedback		110	100		80	60.6	MHz
$f_{MAX}^{(8)}$	Max. Frequency, $1/t_{CP}$ — No Feedback (Pipelined)		110	100		80	62.5	MHz
$t_{SU}$	Input or I/O Setup Time to CLK	7			10			ns
$t_H$	Input or I/O Hold Time from CLK	0			0			ns
$t_{CO1}^{(8)}$	CLK High to Output Valid	3		6.5	3		7	ns
$t_{CO2}$	CLK High to Output Valid Fed through Combinatorial Macrocell	4.5		11	4.5		13	ns
$t_{CNT}^{(8)}$	Macrocell Output Feedback to Macrocell Input—Internal Path	10			16.5			ns
$t_{CL}$	CLK Low Time	4			7			ns
$t_{CH}$	CLK High Time	4			7			ns
$t_{CP}$	CLK Period	10			16			ns

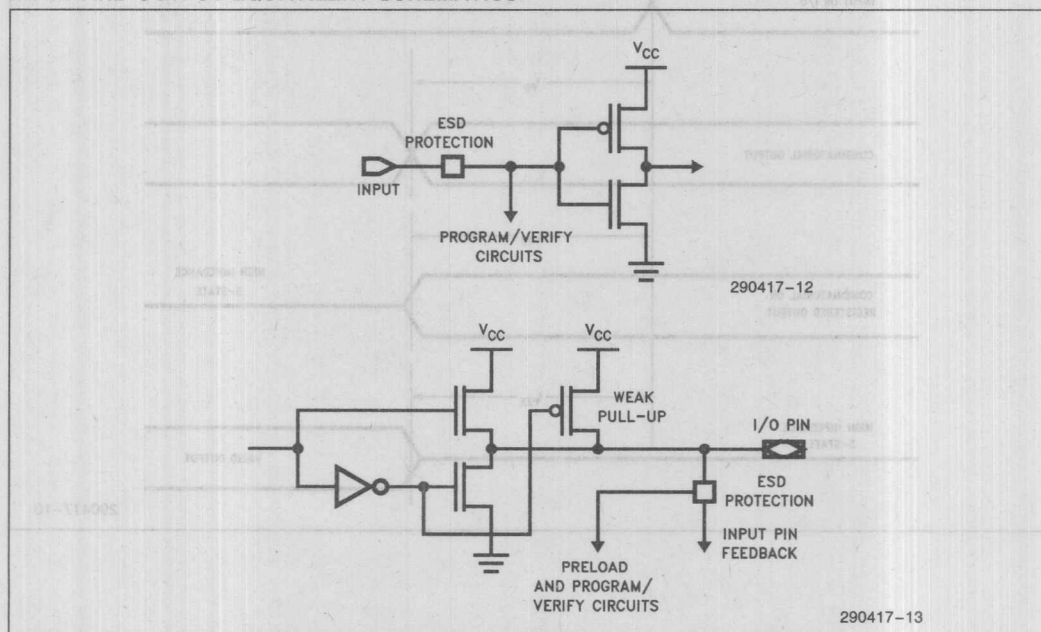
2

**COMBINATORIAL MODE**

## REGISTERED MODE



## INPUT AND OUTPUT EQUIVALENT SCHEMATICS





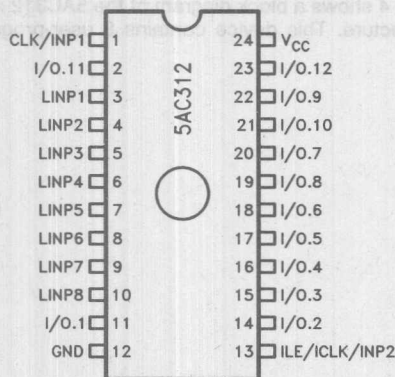
# 5AC312

## 1-MICRON CHMOS 12-MACROCELL EPLD

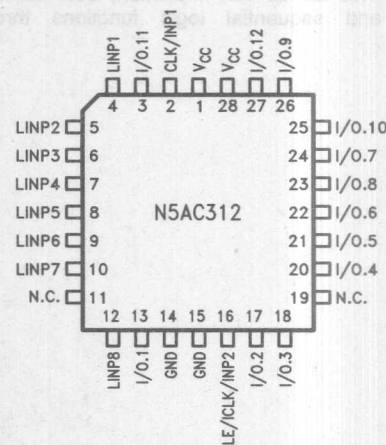
- High-Performance LSI Semi-Custom Logic Alternative for Low-End Gate Arrays, TTL, and 74HC- or 74HCT SSI and MSI Logic, and PLDs
- High Speed  $t_{PD}$  25 ns, 66 MHz Performance Pipelined, 33.3 MHz w/Feedback
- 12 Macrocells with Programmable I/O Architecture; Up To 22 Inputs (10 Dedicated, 12 I/O)
- 8 Programmable Inputs Configurable as Latches, Registers, or Flow-Through
- Flow-Through Input or Global CLK Pin; 1 Flow-Through Input or Global ILE/ICLK Pin
- Programmable AND, Allocatable OR Design Allows up to 16 P-Terms per Macrocell
- Software-Supported P-Term Allocation Between Adjacent Macrocells
- Programmable Output Registers Configurable as D, T, JK, or SR Types
- Dual Feedback on All Macrocells for Implementing Buried Registers with Bidirectional I/O
- 2 P-Terms on All Macrocell Control Signals
- Programmable Low-Power Option for Standby Operation; 100  $\mu$ A Typical Standby Current
- UV Erasable (CerDIP) EPROM Technology or OTP
- 100% Generically Tested EPROM Logic Control Array
- Programmable Security Bit Allows 100% Protection of Proprietary Designs
- Available in 24-Pin 300-mil CerDIP/PDIP and 28-Pin PLCC Packages

(See Packaging Spec., Order Number 240800, Package Type D, P and N)

2



290156-1



290156-2

Figure 1. Pin Configurations



## INTRODUCTION

The Intel 5AC312 CHMOS EPLD (Erasable Programmable Logic Device) represents an innovative approach to overcoming the primary limitations of standard PLDs. Due to a proprietary I/O architecture and macrocell structure, the 5AC312 is capable of implementing high performance logic functions more effectively than previously possible. It can be used as an alternative to low-end gate arrays, multiple programmable logic devices or LS-, HC- or HCT SSI and MSI logic devices. Input and macrocell features for the 5AC312 are a superset of features offered by other PLD-type products.

The 5AC312 uses advanced CHMOS EPROM cells as logic control elements instead of poly-silicon fuses. This technology allows the 5AC312 to operate at levels necessary in high performance systems while significantly reducing the power consumption. Its programmable stand-by function reduces power consumption to almost "zero" in applications where a slight speed loss is traded for power savings.

## ARCHITECTURE DESCRIPTION

The architecture of the 5AC312 is based on the familiar "Sum-Of-Products" programmable AND, fixed OR structure, though the 5AC312 macrocell contains a number of significant functional enhancements. This device can implement both combinational and sequential logic functions through

a highly flexible macrocell and I/O structure. The 5AC312 has been designed to effectively implement both combinational-register and register-combinational-register forms of logic to easily accommodate state machine designs.

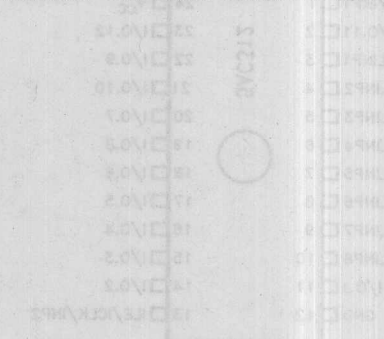
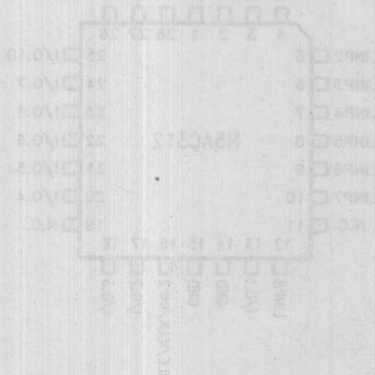
Figure 2 shows a global view of the 5AC312 architecture. The 5AC312 contains a total of 12 I/O macrocells, 8 user-programmable input structures, and 2 additional inputs that can be programmed to serve as either combinatorial inputs or clock inputs. Each of the eight inputs can be individually configured as a latch, register, or flow-through input. Input latches/registers can be synchronously or asynchronously clocked.

Each macrocell is further sub-divided into 16 Product Terms with 8 Product Terms dedicated to the control signals OE, PRESET, ASYNCH. CLK and CLEAR, and 8 Product Terms available for the general data array (see Figure 3).

The basic macrocell architecture of the 5AC312 includes a user-programmable AND array and a user-configurable OR array. The inputs to the programmable AND array originate from the true and complement signals from the programmable input structure, the dedicated inputs, and the 24 feedback paths from the 12 I/O macrocells.

## PROGRAMMABLE INPUTS

Figure 4 shows a block diagram of the 5AC312 input architecture. This device contains 8 user-program-



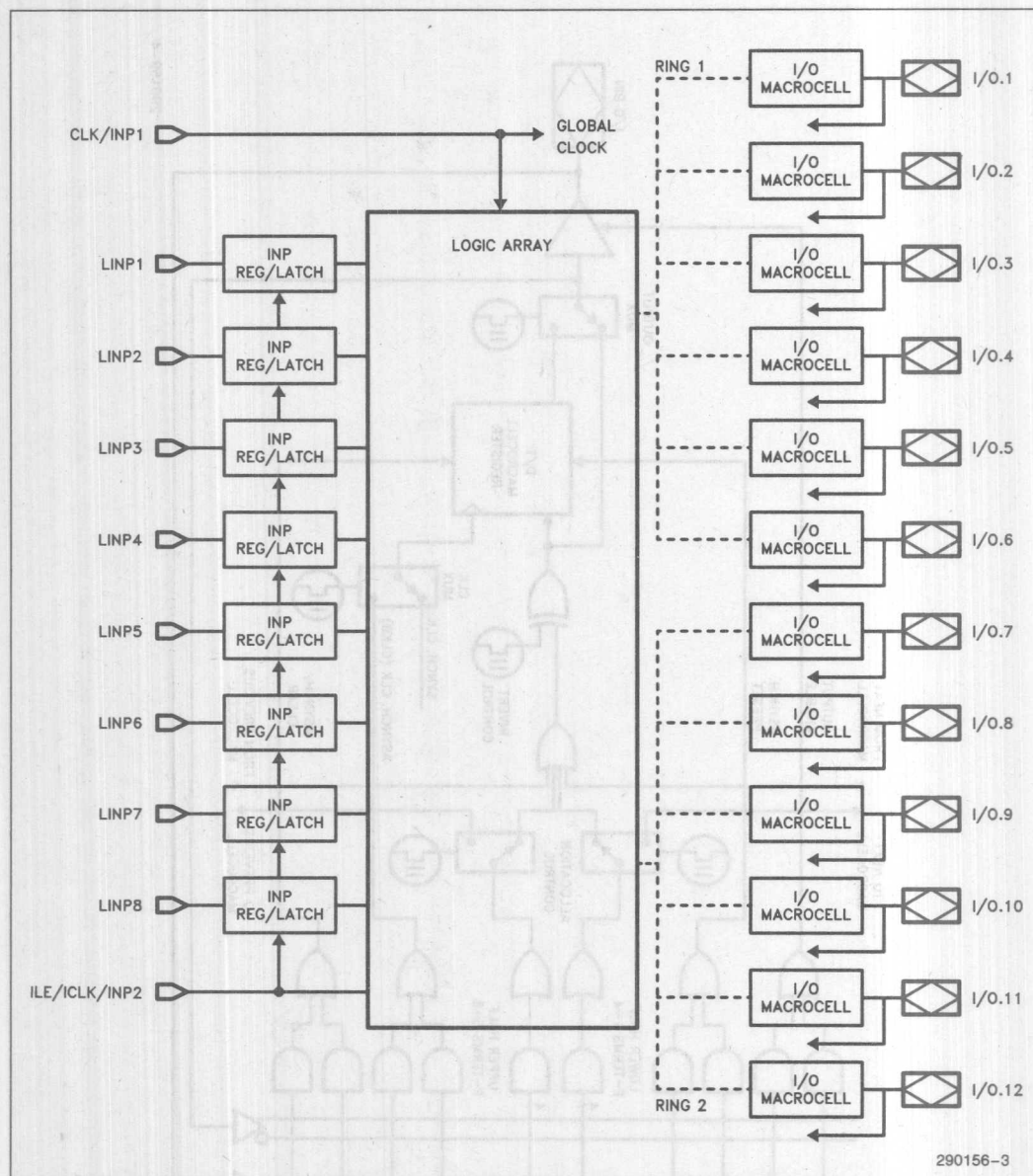
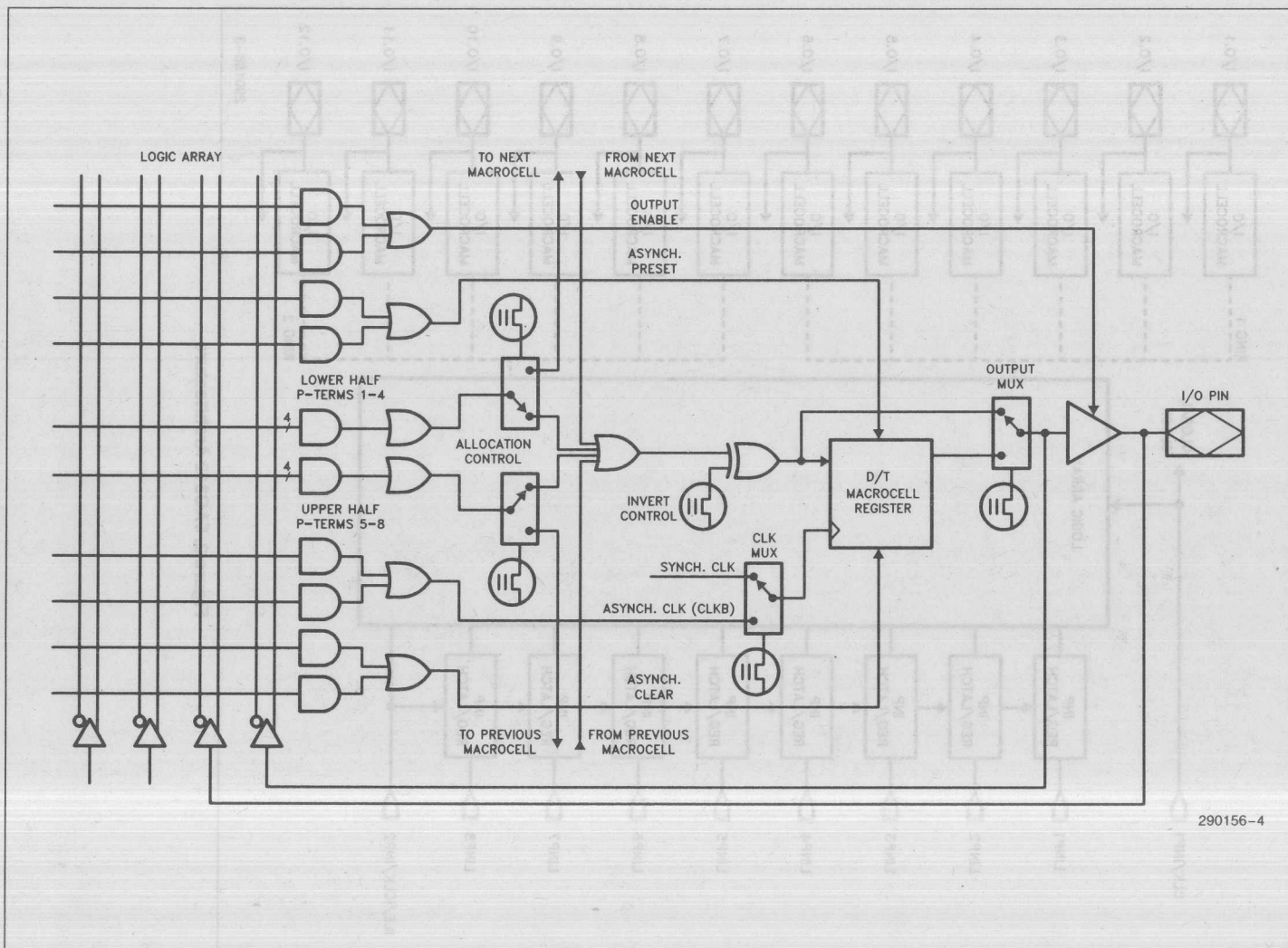


Figure 2. 5AC312 Architecture



290156-4

Figure 3. 5AC312 Basic Macrocell Structure





## MACROCELLS

Each of 12 macrocells in the 5AC312 contains 8 p-terms (Product Terms) to support logic functions. These 8 p-terms are subdivided into 2 groups each containing 4 p-terms. This grouping of p-terms supports the proprietary p-term allocation scheme.

## Register Configuration

Each macrocell can be configured as a D, T, RS, or JK register. The 8 p-terms for control functions are organized so that 2 p-terms support *each of the four* control signals. Control signals in the 5AC312 are: Output Enable (OE), asynchronous I/O register preset (PRESET), asynchronous clock for I/O registers (ASYNCH. CLK), and asynchronous I/O register reset (CLEAR). Availability of 2 p-terms per control signal is another feature that increases the efficiency of the device by reducing the need to use intermediate macrocells sometimes needed to implement control functions.

CLK is a global clock signal that can be used to synchronously clock any or all macrocell registers. It can be used as an input to the logic array at the same time as a macrocell clock. When CLK is not used as a synchronous clock, it functions only as a dedicated input to the logic array.

## Combinatorial Configuration

The macrocell register can be bypassed to implement combinatorial logic functions. When configured to provide combinatorial logic, only the OE control signal is used.

## Invert Select Bit

An invert select EPROM bit is used to invert the product term input into each macrocell register, including double inputs on JK and SR registers. This invert option allows the highest possible logic utilization by use of DeMorgan's logic inversion.

## LOGIC ARRAY

Each intersecting point in the logic array contains a programmable EPROM connection. Initially (erased state), all connections are complete, i.e., both true and complement states of all signals are connected to each p-term.

Connections are opened during programming. When both the true and complement connections exist, a logical false results on the output of the AND gate. If both the true and complement connections of a signal are programmed "open", then a logic "don't care" results for that signal. If all connections for a p-term are programmed open, then a logical true results on the output of the AND gate.

## PRODUCT TERM ALLOCATION

Product Term allocation is defined as taking logic resources (p-terms) away from macrocells where they are not used to support demand for more than 8 Product Terms in other areas of the chip. In the 5AC312, this allocation can occur in increments of 4 p-terms between adjacent macrocells.

The 12 macrocells available in the 5AC312 are grouped into two "rings" with 6 macrocells per ring. Product Terms can be allocated in a "shift register" mode inside a ring; allocation of Product Terms between the rings is not supported. The two rings are shown in Figure 2 and listed in Table 1.

### Example:

The logic function in macrocell 4 requires 16 p-terms. In this case, the iPLS II software allocates 4 p-terms from the previous macrocell in Ring 1 (macrocell 3) and 4 p-terms from the next macrocell in Ring 1 (macrocell 5) to accumulate a total of 16 p-terms ( $8 + 4 + 4$ ). This implementation leaves macrocells 3 and 5 with a remainder of 4 p-terms each. These remaining p-terms in macrocells 3 and 5 can also be allocated away to or can be supplemented with p-terms from their respective previous/next macrocells in Ring 1.

Applying this scheme to the 5AC312 it becomes clear that any macrocell inside the device can support logic functions requiring between 0 and 16 Product Terms. Product Terms allocated away from a macrocell do not affect that macrocell's output structure. If all Product Terms are allocated "away" from a macrocell, the input to that macrocell's I/O control block is tied to GND. This polarity can be changed by programming the invert select EPROM bit. The I/O register as well as all secondary controls to this I/O control block are still available and can be used if needed.

The Product Term allocation scheme described above is automatically supported by iPLS II V2.0 and is transparent to the user. Users can still use explicit pin assignments, but should assign pins in a way that does not conflict with p-term allocation. Software support allows the control signals on macrocells to be used to implement simple logic functions even when all the input p-terms have been allocated to adjacent macrocells.

Table 1. Product Term Allocation Rings

Ring 1			Ring 2		
Current Macro-cell	Next Macro-cell	Previous Macro-cell	Current Macro-cell	Next Macro-cell	Previous Macro-cell
1	2	6	7	8	12
2	3	1	8	9	7
3	4	2	9	10	8
4	5	3	10	11	9
5	6	4	11	12	10
6	1	5	12	7	11

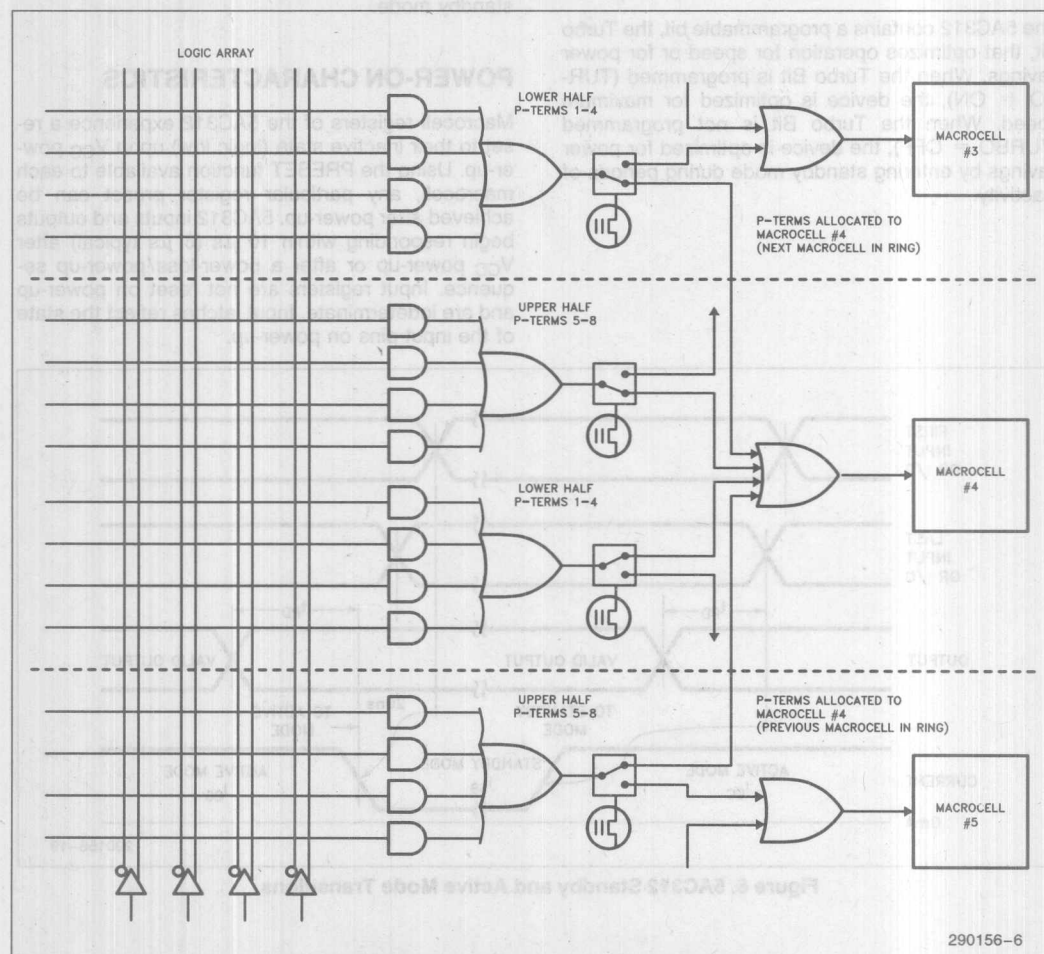


Figure 5. Product Term Allocation (8 + 4 + 4)

## DUAL-FEEDBACK/BURIED LOGIC

Macrocell output can be fed back to the logic array on either one of the two feedback paths. If the pin feedback is used (connected after the output buffer), bidirectional I/O can be implemented. If the internal feedback path is used to implement a buried register or buried logic function, the pin feedback is still available for use as an input. The availability of dual feedbacks on the 5AC312 enhances resource efficiency over single feedback devices.

## AUTOMATIC STANDBY MODE

The 5AC312 contains a programmable bit, the Turbo Bit, that optimizes operation for speed or for power savings. When the Turbo Bit is programmed (TURBO = ON), the device is optimized for maximum speed. When the Turbo Bit is not programmed (TURBO = OFF), the device is optimized for power savings by entering standby mode during periods of inactivity.

Figure 6 shows the device entering standby mode approximately 100 ns after the last input or I/O transition. When the next input or I/O transition is detected, the device returns to active mode. Wakeup time adds an additional 20 ns to the propagation delay through the device as measured from the first transition. No delay will occur if an output is dependent on more than one input and the last of the inputs changes after the device has returned to active mode.

After erasure, the Turbo Bit is unprogrammed (OFF); automatic standby mode is enabled. When the Turbo Bit is programmed (ON), the device never enters standby mode.

## POWER-ON CHARACTERISTICS

Macrocell registers of the 5AC312 experience a reset to their inactive state (logic low) upon  $V_{CC}$  power-up. Using the PRESET function available to each macrocell, any particular register preset can be achieved after power-up. 5AC312 inputs and outputs begin responding within 10  $\mu$ s (6  $\mu$ s typical) after  $V_{CC}$  power-up or after a power-loss/power-up sequence. Input registers are not reset on power-up and are indeterminate. Input latches reflect the state of the input pins on power-up.

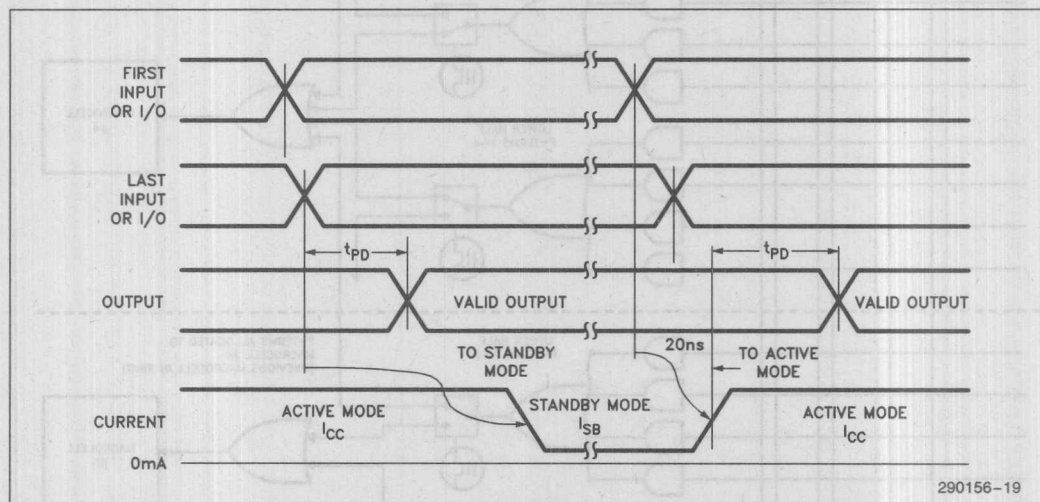


Figure 6. 5AC312 Standby and Active Mode Transitions

## ERASED STATE CONFIGURATION

After erasure and prior to programming, all macro-cells are configured as combinatorial, inverted outputs with output buffers three-stated. Inputs are configured as synchronous registers.

## ERASURE CHARACTERISTICS

Erasure time for the 5AC312 is 1 hour at 12,000  $\mu\text{W}/\text{cm}^2$  with a 2537Å UV lamp.

Erasure characteristics of the device are such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000Å. It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000Å–4000Å range. Data shows that constant exposure to room level fluorescent lighting could erase the typical 5AC312 in approximately six years, while it would take approximately two weeks to erase the device when exposed to direct sunlight. If the device is to be exposed to these lighting conditions for extended periods of time, conductive opaque labels should be placed over the device window to prevent unintentional erasure.

The recommended erasure procedure for the 5AC312 is exposure to shortwave ultraviolet light with a wavelength of 2537Å. The integrated dose (i.e., UV intensity  $\times$  exposure time) for erasure should be a minimum of forty (40) Wsec/cm<sup>2</sup>. The erasure time with this dosage is approximately 1 hour using an ultraviolet lamp with a 12,000  $\mu\text{W}/\text{cm}^2$  power rating. The device should be placed within 1 inch of the lamp tubes during exposure. The maximum integrated dose the 5AC312 can be exposed to without damage is 7258 Wsec/cm<sup>2</sup> (1 week at 12,000  $\mu\text{W}/\text{cm}^2$ ). Exposure to high intensity UV light for longer periods may cause permanent damage to the device.

## Intelligent Programming Algorithm

The 5AC312 supports the Intelligent Programming algorithm which rapidly programs Intel PLDs, while maintaining a high degree of reliability. This method ensures reliability as the incremental program margin of each bit has been verified in the programming process. (Programming information for the 5AC312 is available from Intel by request.)

## DESIGN SECURITY

A Security Bit provides a programmable security option to protect the data programmed in the device. Once this bit is set during programming, subsequent attempts to read the device architecture information are prevented. This method provides a higher degree of design security than fuse-based devices, since programmed EPROM cells are invisible even to microscopic examination. The Security Bit (also called the Verify Protect Bit), along with all the other EPROM cells, is reset by erasing the device.

## LATCH-UP IMMUNITY

All of the input, I/O, and clock pins of the device have been designed to resist latch-up which is inherent in inferior CMOS structures. The 5AC312 is designed with Intel's proprietary 1-micron CHMOS EPROM process. Thus, each of the pins will not experience latch-up with currents up to  $\pm 100$  mA and voltages ranging from  $-0.5\text{V}$  to  $(V_{CC} + 0.5\text{V})$ . The programming pin is designed to resist latch-up to the 13.5 maximum device limit.

## DESIGN RECOMMENDATIONS

For proper operation, it is recommended that all input and output pins be constrained to the voltage range  $(\text{GND} < (V_{IN} \text{ or } V_{OUT}) < V_{CC})$ . All unused inputs and I/Os should be tied to  $V_{CC}$  or  $\text{GND}$  to minimize power consumption (do not leave them floating). A power supply decoupling capacitor of at least 0.2  $\mu\text{F}$  must be connected directly between each  $V_{CC}$  and  $\text{GND}$  pin.

As with all CMOS devices, ESD handling procedures should be used with the 5AC312 to prevent damage to the device during programming, assembly, and test.

## FUNCTIONAL TESTING

Since the logical operation of the 5AC312 is controlled by EPROM elements, the device is completely testable during the manufacturing process. Each programmable EPROM bit controlling the internal logic is tested using application-independent test patterns. EPROM cells in the 5AC312 are 100% tested for programming and erase. After testing, the devices are erased before shipments to the customers. No post-programming tests of the EPROM array are required.



The testability and reliability of EPROM-based programmable logic devices are important features over similar devices based on fuse technology. Fuse-based programmable logic devices require a user to perform post-programming tests to insure device functionality. During the manufacturing process, tests on these parts can only be performed in very restricted manners to prevent pre-programming of the array.

## INTEL PROGRAMMABLE LOGIC SOFTWARE SUPPORT

Full logic compilation and functional simulation for the 5AC312 is supported by PLDshell Plus™ software. The GUI Logic-IID provides programming support on Intel programmers.

PLDshell Plus design software is Intel's new, user-friendly design tool for  $\mu$ PLD design. PLDshell Plus allows users to incorporate their preferred text editor, programming software, and additional design tools into a easy-to-use, menued design environment that includes Intel's PLDasm™ logic compiler and simulation software along with disassembly, conversion and translation utilities. The PLDasm compiler and simulator software accepts industry-standard PDS source files that express designs as Boolean equations, truth tables, or state machines. On-line help, datasheet briefs, technical notes, and error message information, along with waveform viewing/printing capability make the design task as easy as possible. PLDshell Plus software is available from Intel Literature channels or from your local Intel sales representative.

Tools that support schematic capture and timing simulation for the 5AC312 are available. Support under iPLS II is still available. Please refer to the "Development Tools" section of the Programmable Logic handbook.

The 5AC312 is also supported by third-party logic compilers such as ABEL\*, CUPL\*, PLDesigner\*, Log/IC\*, etc. Programming support is provided by third-party programmer companies such as Data I/O, Logical Devices, STAG, etc. Please refer to the "Third-Party Support" lists in the *Programmable Logic* handbook for complete information and vendor contacts.

## ORDERING INFORMATION

t <sub>PD</sub> (ns)	t <sub>CO</sub> (ns)	f <sub>MAX</sub> (MHz)	Order Code	Package	Operating Range
25	15	66	D5AC312-25	†CERDIP	Commercial
			P5AC312-25	PDIP	
			N5AC312-25	PLCC	
30	18	50	D5AC312-30	†CERDIP	Commercial
			P5AC312-30	PDIP	
			N5AC312-30	PLCC	

†Windowed package allows UV erase.

PLDshell Plus™ is a trademark of Intel Corporation

\*ABEL is a trademark of Data I/O, Corporation. CUPL is a trademark of Logical Devices, Inc. PLDesigner is a trademark of MINC, Inc. Log/IC is a trademark of ISDATA, Inc.

**ABSOLUTE MAXIMUM RATINGS\***Supply Voltage ( $V_{CC}$ ) (1) ..... -2.0V to +7.0V

Programming Supply

Voltage ( $V_{PP}$ ) (1) ..... -2.0V to +13.5VD.C. Input Voltage ( $V_I$ ) (1, 2) ... -0.5V to  $V_{CC} + 0.5V$ Storage Temperature ( $T_{stg}$ ) ..... -65°C to +150°CAmbient Temperature ( $T_{amb}$ ) (3) .. -10°C to +85°C**NOTES:**

1. Voltages with respect to GND.

2. Minimum D.C. input is -0.5V. During transitions, the inputs may undershoot to -2.0V or overshoot to +7V for periods of less than 20 ns under no load conditions.

3. Under bias. Extended temperature range versions are available.

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

*\*WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.***RECOMMENDED OPERATING CONDITIONS**

Symbol	Parameter	Min	Max	Unit
$V_{CC}$	Supply Voltage	4.75	5.25	V
$V_{IN}$	Input Voltage	0	$V_{CC}$	V
$V_O$	Output Voltage	0	$V_{CC}$	V
$T_A$	Operating Temperature	0	+70	°C
$t_R$	Input Rise Time		500	ns
$t_F$	Input Fall Time		500	ns

2

**PACKAGE/TECHNOLOGY SPECIFICATIONS**

Description	Specification
$\Theta_{Ja}$ —Junction-to-Ambient Thermal Resistance	47°C/W—CerDIP 47°C/W—PDIP 47°C/W—PLCC
$\Theta_{Jc}$ —Junction-to-Case Thermal Resistance	16°C/W—CerDIP 16°C/W—PDIP 16°C/W—PLCC
$I_{CC}$ Hot—Ambient @70°C	80 mA
$I_{CC}$ Typical—Ambient @25°C	80 mA
Process	CHMOS IIIE, PX29

**D.C. CHARACTERISTICS**  $T_A = 0^\circ\text{C to } +70^\circ\text{C}$ ,  $V_{CC} = 5.0V \pm 5\%$ 

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
$V_{IH}^{(4)}$	High Level Input Voltage	2.0		$V_{CC} + 0.3$	V	
$V_{IL}^{(4)}$	Low Level Input Voltage	-0.3		0.8	V	
$V_{OH}^{(5)}$	High Level Output Voltage	2.4			V	$I_O = -4.0$ mA D.C., $V_{CC} = \text{min.}$
$V_{OL}$	Low Level Output Voltage			0.45	V	$I_O = 8.0$ mA D.C., $V_{CC} = \text{min.}$
$I_I$	Input Leakage Current			$\pm 10$	$\mu\text{A}$	$V_{CC} = \text{max.}$ , $\text{GND} < V_{IN} < V_{CC}$
$I_{OZ}$	Output Leakage Current			$\pm 10$	$\mu\text{A}$	$V_{CC} = \text{max.}$ , $\text{GND} < V_{OUT} < V_{CC}$
$I_{SC}^{(6)}$	Output Short Circuit Current	-30		-90	mA	$V_{CC} = \text{max.}$ , $V_{OUT} = 0.5V$
$I_{SB}^{(7)}$	Standby Current		100	300	$\mu\text{A}$	$V_{CC} = \text{max.}$ , $V_{IN} = V_{CC}$ or GND, Standby Mode

# D.C. CHARACTERISTICS

$T_A = 0^\circ\text{C to } +70^\circ\text{C}$ ,  $V_{CC} = 5.0\text{V} \pm 5\%$  (Continued)

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
$I_{CC}$	Power Supply Current (See $I_{CC}$ vs Freq. Graph)		10		mA	$V_{CC} = \text{max.}$ , $V_{IN} = V_{CC}$ or GND, No Load, Input Freq. = 1 MHz Active Mode (Turbo = Off), Device Prog. as 12-Bit Ctr.

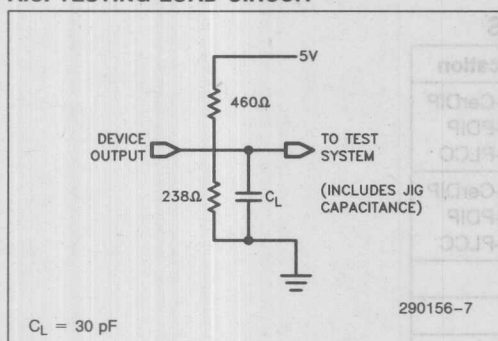
## NOTES:

4. Absolute values with respect to device GND; all over and undershoots due to system or tester noise are included. Do not attempt to test these values without suitable equipment.
5.  $I_O$  at CMOS levels (3.84V) = -2 mA.
6. Not more than 1 output should be tested at a time. Duration of that test must not exceed 1 second.
7. With Turbo Bit Off, device automatically enters standby mode approximately 100 ns after last input transition.

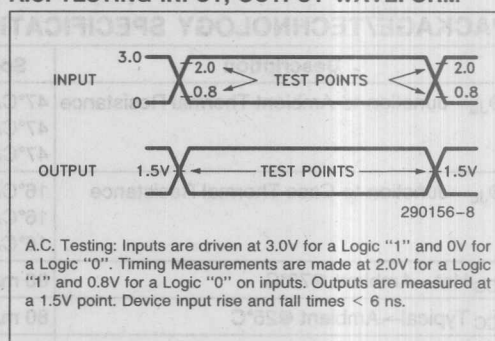
## CAPACITANCE

Symbol	Parameter	Min	Typ	Max	Unit	Conditions
$C_{IN}$	Input Capacitance			8	pF	$V_{IN} = 0\text{V}$ , $f = 1.0\text{ MHz}$
$C_{OUT}$	I/O Capacitance			15	pF	$V_{OUT} = 0\text{V}$ , $f = 1.0\text{ MHz}$
$C_{CLK}$	ILE/ICLK/INP2 Capacitance			12	pF	$V_{IN} = 0\text{V}$ , $f = 1.0\text{ MHz}$
$C_{VPP}$	$V_{PP}$ Pin (CLK/INP1)			25	pF	$V_{IN} = 0\text{V}$ , $f = 1.0\text{ MHz}$

## A.C. TESTING LOAD CIRCUIT



## A.C. TESTING INPUT, OUTPUT WAVEFORM



# A.C. CHARACTERISTICS

$T_A = 0^\circ\text{C to } +70^\circ\text{C}$ ,  $V_{CC} = 5.0\text{V} \pm 5\%$ , Turbo Bit "On"<sup>(8)</sup>

Symbol	From	To	5AC312-25			5AC312-30			Non-(10) Turbo Mode	Unit
			Min	Typ	Max	Min	Typ	Max		
$t_{PD}$	Input or I/O	Comb. Output		20	25		25	30	+ 20	ns
$t_{PZX}^{(9)}$	Input or I/O	Output Enable		20	25		25	30	+ 20	ns
$t_{PXZ}^{(9)}$	Input or I/O	Output Disable		20	25		25	30	+ 20	ns
$t_{CLR}$	Asynch. Reset	Q Reset		20	25		25	30	+ 20	ns
$t_{SET}$	Asynch. Set	Q Set		20	25		25	30	+ 20	ns

## NOTES:

8. Typical values are at  $T_A = 25^\circ\text{C}$ ,  $V_{CC} = 5\text{V}$ , Active Mode.
9.  $t_{PZX}$  and  $t_{PXZ}$  are measured at  $\pm 0.5\text{V}$  from steady-state voltage as driven by spec. output load.  $t_{PXZ}$  is measured with  $C_L = 5\text{ pF}$ .
10. If device is operated with Turbo Bit Off (Non-Turbo Mode) and the device has been inactive for approximately 100 ns, increase time by amount shown.

**SYNCHRONOUS CLOCK MODE (MACROCELLS) A.C. CHARACTERISTICS**T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5.0V ±5%, Turbo Bit On<sup>(8)</sup>

Symbol	Parameter	5AC312-25			5AC312-30			Non-(10) Turbo Mode	Unit
		Min	Typ	Max	Min	Typ	Max		
f <sub>MAX</sub>	Max. Frequency (Pipelined) 1/t <sub>SU</sub> —No Feedback		80	66		66	50		MHz
f <sub>CNT1</sub>	Max. Count Frequency 1/(t <sub>SU</sub> + t <sub>CO</sub> )—External Feedback		40	33.3		33	26.3		MHz
f <sub>CNT2</sub>	Max. Count Frequency 1/t <sub>CNT</sub> —Internal Feedback		40	33.3		35	28.5		MHz
t <sub>SU1</sub>	Input Setup Time to CLK ↑	15	12		20	18		+20	ns
t <sub>SU2</sub>	I/O Setup Time to CLK ↑	15	12		20	18		+20	ns
t <sub>H</sub>	I or I/O Hold after CLK ↑	0			0				ns
t <sub>CO</sub>	CLK ↑ to Output Valid		10	15		12	18		ns
t <sub>CNT</sub>	Macrocell Output Feedback to Macrocell Input—Internal Path		25	30		30	35	+20	ns
t <sub>CH</sub>	CLK High Time	7			9				ns
t <sub>CL</sub>	CLK Low Time	7			9				ns
t <sub>CW</sub>	Minimum Clock Period	15			20				ns

2

**SYNCHRONOUS CLOCK MODE (INPUT STRUCTURE) A.C. CHARACTERISTICS**T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5.0V ±5%, Turbo Bit On<sup>(8)</sup>

Symbol	Parameter	5AC312-25			5AC312-30			Non-(10) Turbo Mode	Unit
		Min	Typ	Max	Min	Typ	Max		
f <sub>MAXI</sub>	Max. Frequency (1/t <sub>CWI</sub> )		80	66		66	50		MHz
t <sub>SUIR</sub>	Input Register/Latch Setup Time before ILE/ICLK ↓	5			5				ns
t <sub>ESUI</sub> <sup>(11)</sup>	Input Latch Setup Time before ILE ↑	5			5				ns
t <sub>COI</sub>	ICLK ↓ to Comb. Output		30	35		35	40	+20	ns
t <sub>EOI</sub>	ILE ↑ to Comb. Output		30	35		35	40	+20	ns
t <sub>HI</sub>	Input Hold after ICLK/ILE ↓	7			10				ns
t <sub>EHI</sub>	Input Hold after ILE ↓	7			10				ns
t <sub>CHI</sub>	ILE/ICLK High Time	7			9				ns
t <sub>CLI</sub>	ILE/ICLK Low Time	7			9				ns
t <sub>CWI</sub>	Minimum Input Clock Period	15			20				ns

**NOTE:**11. This specification must be met to guarantee t<sub>EOI</sub>. When ILE goes high before data is valid, use t<sub>pD</sub> instead of t<sub>EOI</sub>.



**ASYNCHRONOUS CLOCK MODE INPUT STRUCTURE A.C. CHARACTERISTICS**T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5.0V ±5%, Turbo Bit On<sup>(8)</sup>

Symbol	Parameter	5AC312-25			5AC312-30			Non-(10) Turbo Mode	Unit
		Min	Typ	Max	Min	Typ	Max		
f <sub>AMAXI</sub>	Max. Frequency Input Register 1/(t <sub>ACLI</sub> + t <sub>ACHI</sub> )		80	66		66	50		MHz
t <sub>ASUIR</sub>	Input Register/Latch Setup Time to Asynch. ILE/ICLK	0			0				ns
t <sub>AESUI</sub> <sup>(11)</sup>	Input Latch Setup Time before Asynch. ILE	0			0				ns
t <sub>ACOI</sub>	Asynch. ICLK to Comb. Output		40	48		45	55	+20	ns
t <sub>AEOI</sub>	Asynch. ILE ↑ to Comb. Output		40	48		45	55	+20	ns
t <sub>AHI</sub>	Input Register/Latch Hold after Asynch. ILE/ICLK	20	14		25	20		+20	ns
t <sub>AHI</sub>	Input Hold after Asynch. ILE	20	14		25	20			ns
t <sub>ACHI</sub>	Asynch. ICLK High Time	7			9			+20	ns
t <sub>ACLI</sub>	Asynch. ICLK Low Time	7			9			+20	ns
t <sub>ACWI</sub>	Minimum Input Clock Period	15			20			+20	ns

**ASYNCHRONOUS CLOCK MODE MACROCELLS A.C. CHARACTERISTICS**T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5.0V ±5%, Turbo Bit On<sup>(8)</sup>

Symbol	Parameter	5AC312-25			5AC312-30			Non-(10) Turbo Mode	Unit
		Min	Typ	Max	Min	Typ	Max		
f <sub>AMAX</sub>	Max. Frequency (Pipelined) 1/(t <sub>ACL</sub> + t <sub>ACH</sub> )—No Feedback		80	66		66	50		MHz
f <sub>ACNT1</sub>	Max. Frequency 1/(t <sub>ASU</sub> + t <sub>ACO</sub> )—External Feedback		35	28.5		33	23.8		MHz
f <sub>ACNT2</sub>	Max. Frequency 1/t <sub>ACNT</sub> —with Feedback		40	33.3		35	30		MHz
t <sub>ASU1</sub>	Input Setup Time to Asynch. Clock	10			12			+20	ns
t <sub>ASU2</sub>	I/O Setup Time to Asynch. Clock	10			12			+20	ns
t <sub>AH</sub>	Input or I/O Hold after Asynch. Clock	5	0		5	0			ns
t <sub>ACO</sub>	Asynch. CLK to Output Valid		20	25		25	30	+20	ns
t <sub>ACNT</sub>	Register Output Feedback to Register Input— Internal Path		25	30		30	35	+20	ns
t <sub>ACH</sub>	Asynch. CLK High Time	7			9			+20	ns
t <sub>ACL</sub>	Asynch. CLK Low Time	7			9			+20	ns
t <sub>ACW</sub>	Minimum Asynch. CLK Period	15			20			+20	ns

## INPUT-CLOCK-TO-MACROCELL-CLOCK A.C. CHARACTERISTICS

 $T_A = 0^\circ\text{C to } +70^\circ\text{C}$ ,  $V_{CC} = 5.0\text{V} \pm 5\%$ , Turbo Bit On<sup>(8)</sup>

Symbol	Parameter	5AC312-25			5AC312-30			Non-(10) Turbo Mode	Unit
		Min	Typ	Max	Min	Typ	Max		
$t_{C1C2}^{(12)}$	Synchronous ILE/ICLK to Synchronous Macrocell CLK	25			30			+ 20	ns
	Synchronous ILE/ICLK to Asynchronous Macrocell CLK	15			18			+ 20	ns
	Asynchronous ILE/ICLK to Synchronous Macrocell CLK	35			40			+ 20	ns
	Asynchronous ILE/ICLK to Asynchronous Macrocell CLK	25			35			+ 20	ns

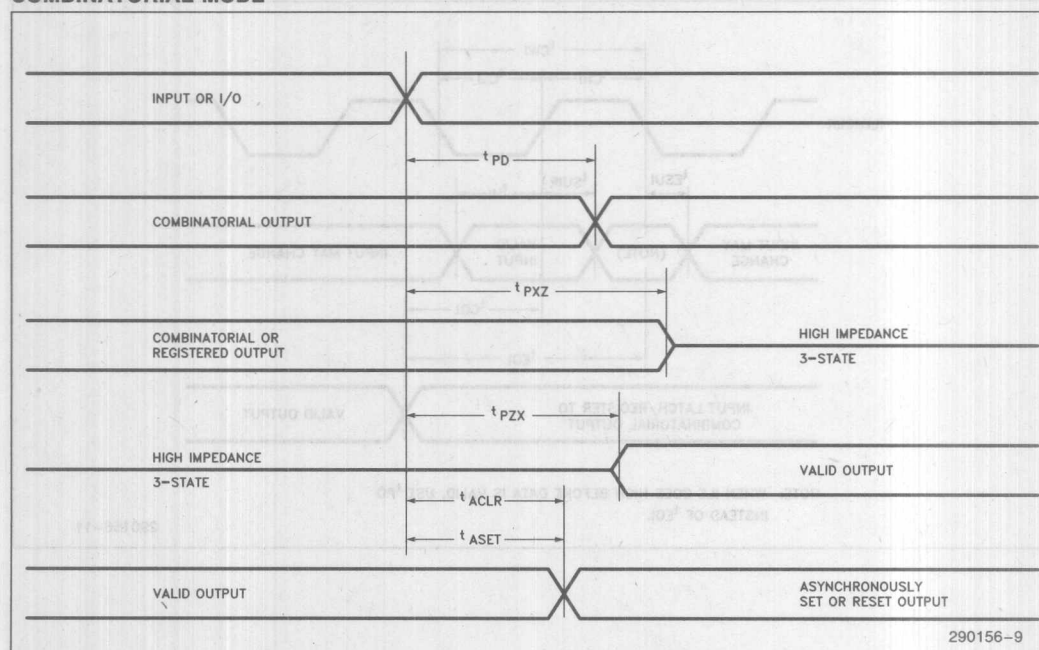
## NOTE:

12. Times for SETUP, HOLD, and OUTPUT VALID are shown in previous tables.

2

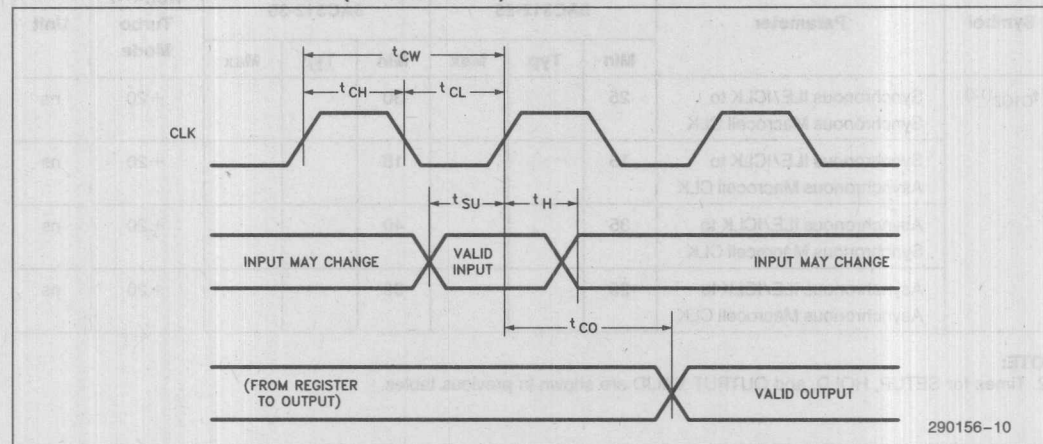
## SWITCHING WAVEFORMS

## COMBINATORIAL MODE

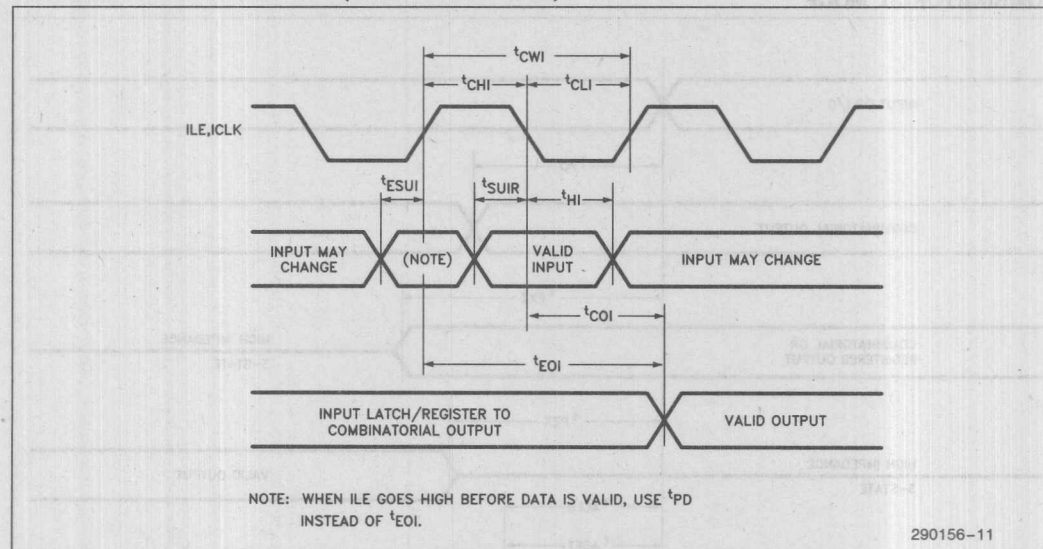


## SWITCHING WAVEFORMS (Continued)

### SYNCHRONOUS CLOCK MODE (MACROCELLS)

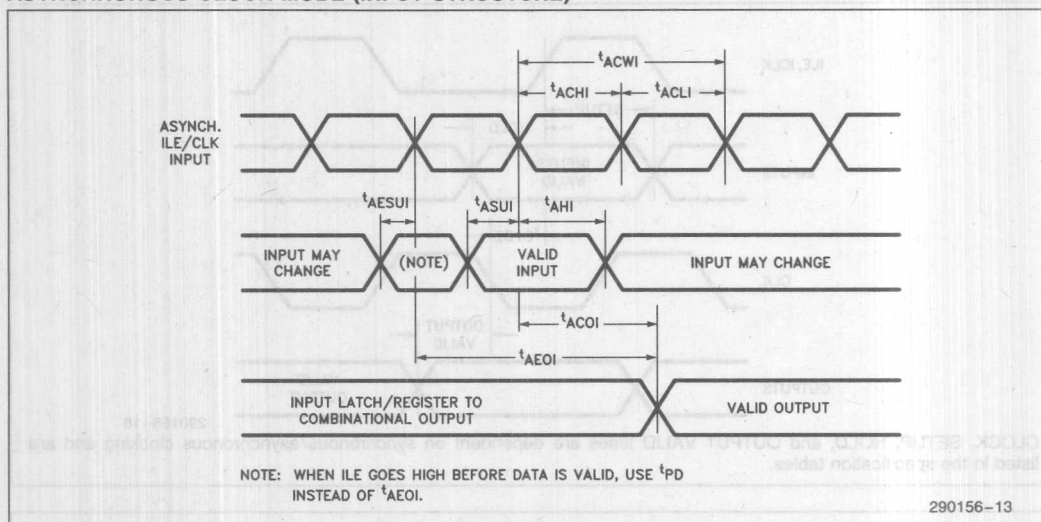


### SYNCHRONOUS CLOCK MODE (INPUT STRUCTURE)

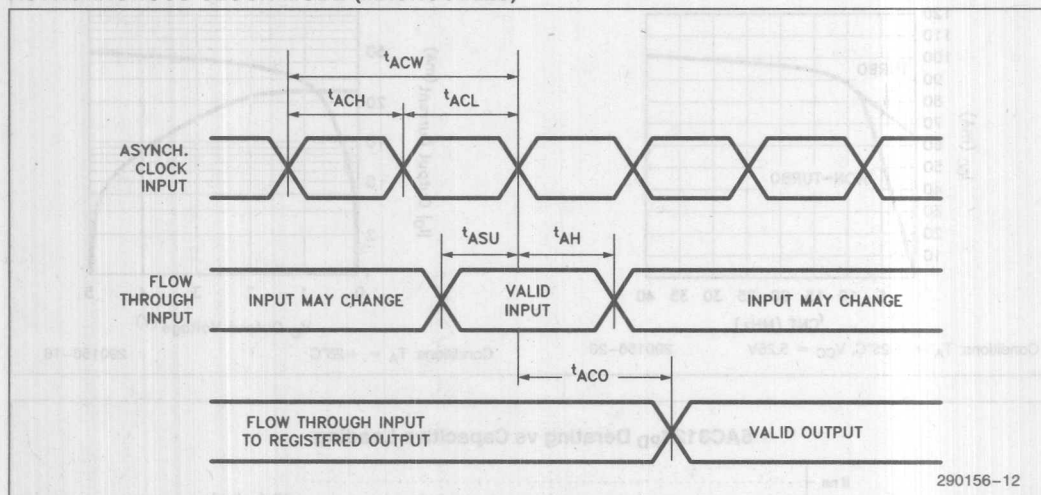


# SWITCHING WAVEFORMS (Continued)

## ASYNCHRONOUS CLOCK MODE (INPUT STRUCTURE)



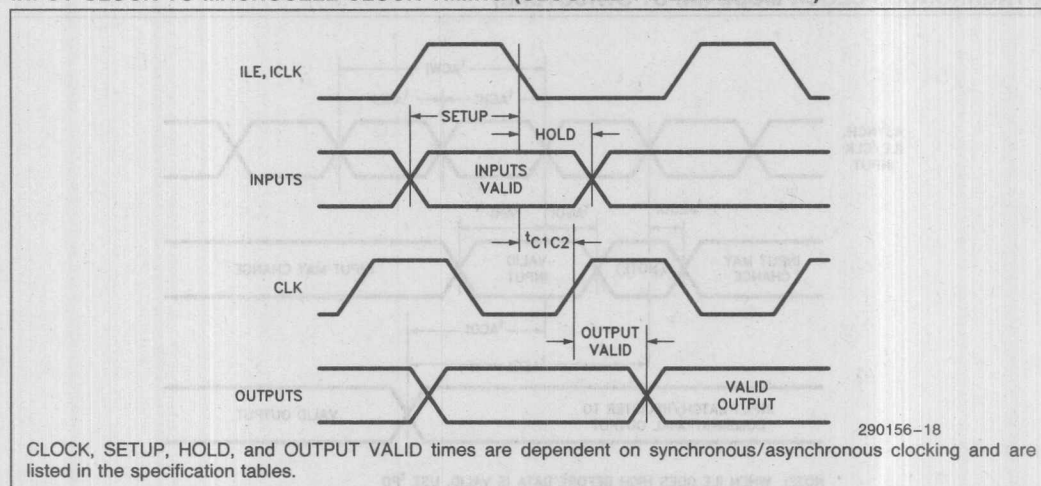
## ASYNCHRONOUS CLOCK MODE (MACROCELLS)



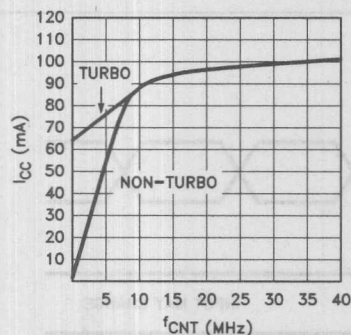


# SWITCHING WAVEFORMS (Continued)

## INPUT CLOCK-TO-MACROCELL CLOCK TIMING (CLOCKED PIPELINED DATA)

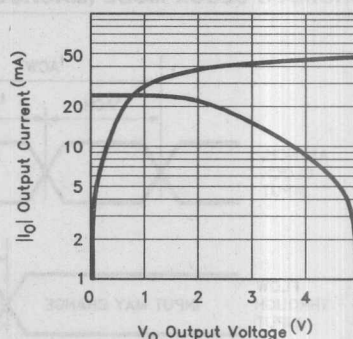


5AC312 Current in Relation to Frequency



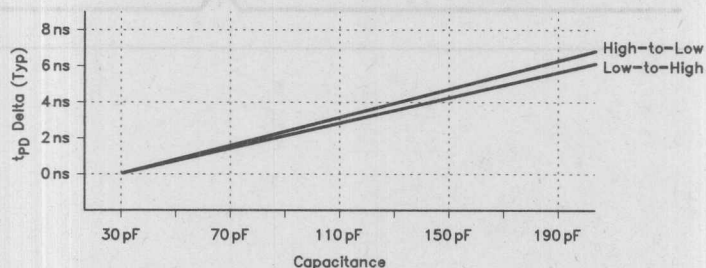
Conditions:  $T_A = +25^\circ\text{C}$ ,  $V_{CC} = 5.25\text{V}$  290156-20

5AC312 Output Drive Current in Relation to Voltage



Conditions:  $T_A = +25^\circ\text{C}$  290156-16

5AC312  $t_{PD}$  Derating vs Capacitive Loading



$T_A = +25^\circ\text{C}$   
 $V_{CC} = 5.0\text{V}$

290156-21

# 5AC324

## 1-MICRON CHMOS 24-MACROCELL EPLD

- High-Performance LSI Semi-Custom Logic Alternative to Low-end Gate Arrays, TTL, 74HC SSI and MSI Logic, and PLDs
  - High Speed  $t_{PD}$  25 ns, 66 MHz Performance Pipelined, 33.3 MHz w/ Feedback
  - 24 Macrocells with Programmable I/O Architecture; Up to 36 Inputs (12 Dedicated, 24 I/O)
  - 10 Programmable Inputs Configurable as Latches, Registers, or Flow-Through
  - 1 Flow-Through Input or Global CLK Pin; 1 Flow-Through Input or Global ILE/ICLK Pin
  - Programmable AND, Allocatable OR Design Allows up to 16 P-Terms per Macrocell
  - Software-Supported P-Term Allocation Between Adjacent Macrocells
  - Programmable Output Registers Configurable as D, T, JK, or SR Types
  - Dual Feedback on All Macrocells for Implementing Buried Registers with Bidirectional I/O
  - 2 P-Terms on All Macrocell Control Signals
  - Programmable Low-Power Option for "Stand-by" Operation; 150  $\mu$ A Typical Standby Current
  - UV Erasable (CerDIP) EPROM Technology or OTP
  - 100% Generically Tested EPROM Logic Control Array
  - JEDEC Pinout
  - Available in 40-pin CerDIP/PDIP and 44-Pin PLCC Packages
- (See Packaging Spec., Order Number 240800, Package Type D, P, and N)

2

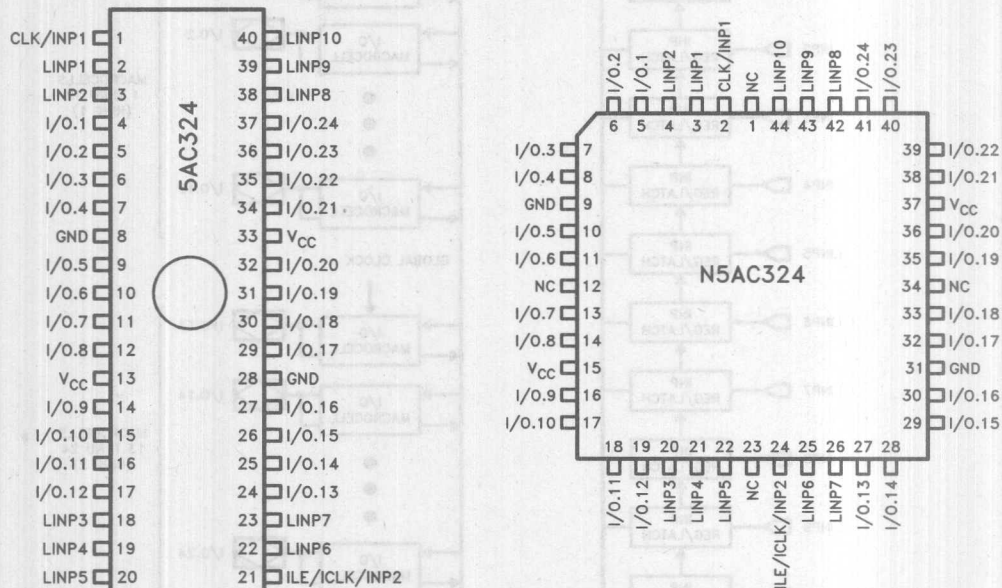


Figure 1. 5AC324 Pinout Diagrams

## INTRODUCTION

The Intel 5AC324 CHMOS EPLD (Erasable Programmable Logic Device) is a high integration device that overcomes the primary limitations of standard PLDs. Due to a proprietary I/O architecture and macrocell structure, the 5AC324 is capable of implementing high performance logic functions more effectively than previously possible. The 5AC324 can be used as an alternative to low-end gate arrays, multiple programmable logic devices, or LS-, HC-, or HCT SSI and MSI logic devices. Input and macrocell features for the 5AC324 are a superset of features offered on other PLD-type products.

The 5AC324 uses advanced CHMOS EPROM cells as logic control elements instead of poly-silicon fuses. This technology allows the device to operate at levels necessary in high performance systems while significantly reducing power consumption. Its programmable standby mode reduces power to near zero in applications where a slight speed loss is traded for power savings.

## ARCHITECTURE DESCRIPTION

The architecture of the 5AC324 is based on the familiar "Sum-Of-Products" programmable AND, fixed OR structure. This structure is then surrounded by powerful, programmable macrocells and inputs. The 5AC324 can implement both combinatorial and sequential logic functions through a highly flexible macrocell and I/O structure. The architecture of the device supports both combinatorial-register and register-combinatorial-register forms of logic to easily accommodate state machine designs.

Figure 2 shows a global view of the 5AC324 architecture. The 5AC324 contains a total of 24 I/O programmable macrocells, 10 programmable input structures, and two clock inputs that can be programmed to function either as combinatorial inputs or clock inputs for the input structures and macrocells.

Each of the ten programmable inputs can be individually configured as a latch, register or flow-through

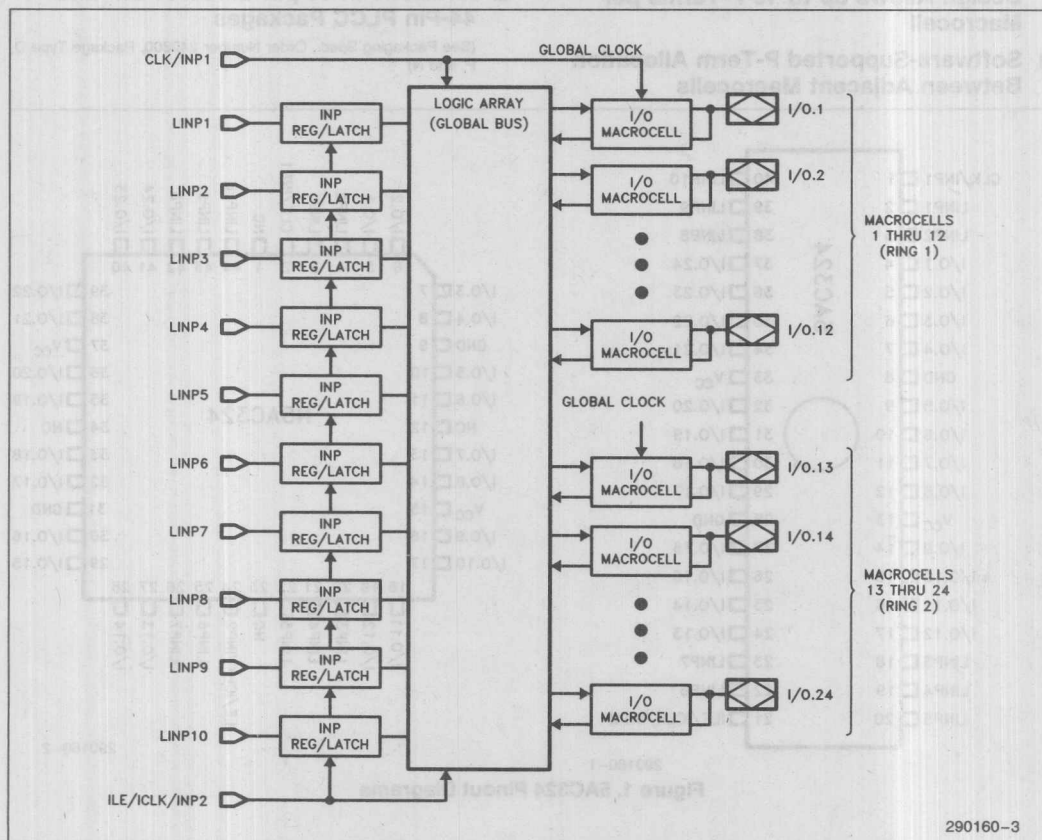
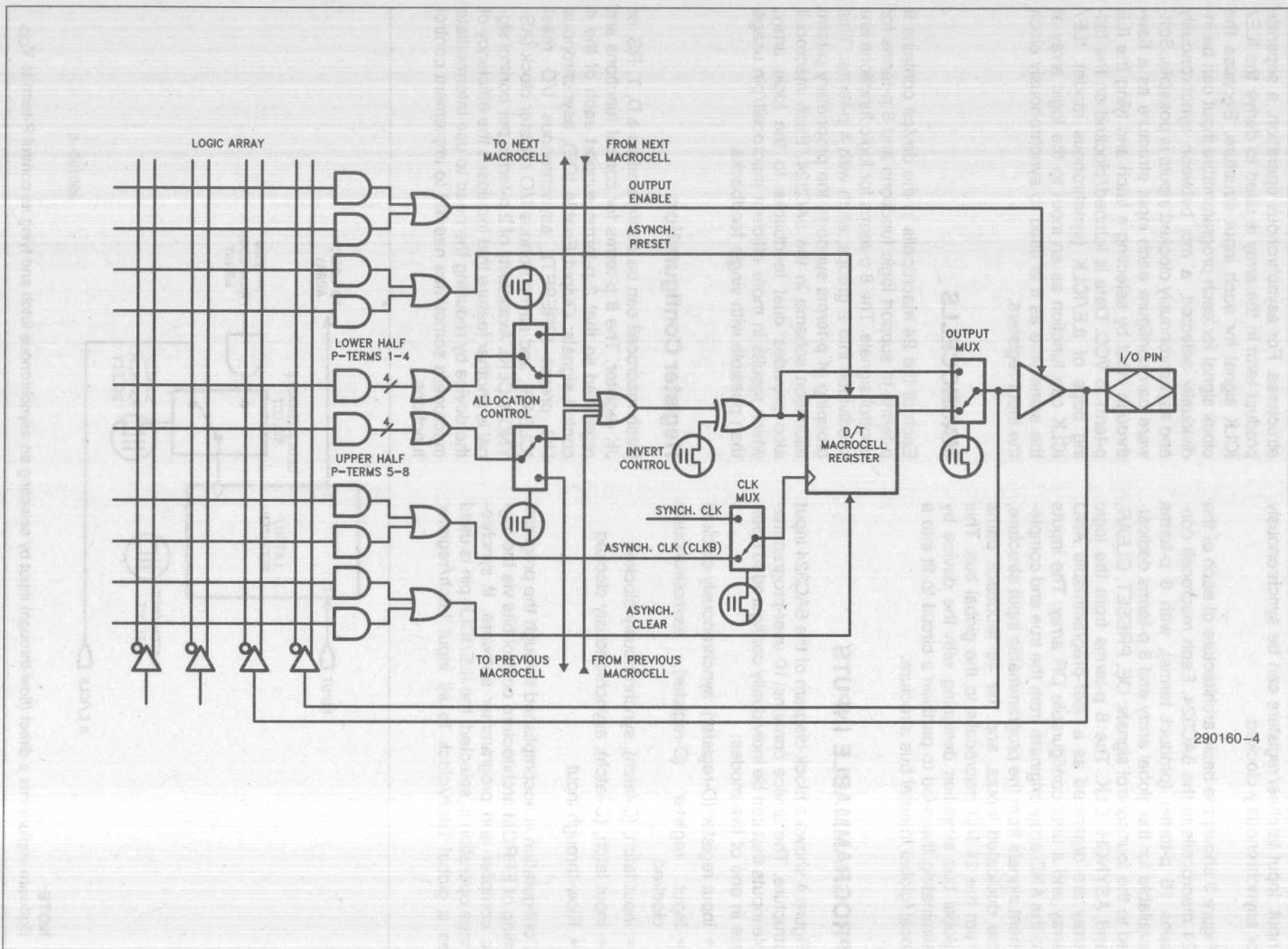


Figure 2. 5AC324 Global Architecture



**Figure 3. 5AC324 Macrocell Architecture**



input. Input latches/registers can be synchronously or asynchronously clocked.

Figure 3 shows the basic architecture of each of the 24 macrocells in the 5AC324. Each macrocell contains 16 p-terms (product terms), with 8 p-terms available for the global array and 8 p-terms dedicated to the four control signals: OE, PRESET, CLEAR, and ASYNCH. CLK. The 8 p-terms from the logic array are organized as a user-programmable AND array and a user-configurable OR array. The inputs to the AND array originate from the true and complement signals from the programmable input structure, the dedicated inputs, and the 48 feedback paths from the 24 I/O macrocells to the global bus. This global bus simplifies designing with the device by eliminating the need to partition a circuit to fit into a local/global internal bus structure.

## PROGRAMMABLE INPUTS

Figure 4 shows a block diagram of the 5AC324 input structure. The device contains 10 user-programmable inputs that can be individually configured to operate in one of five modes:

- input register (D-register), synchronously clocked
- input register (D-register), asynchronously clocked
- input latch, (D-latch), synchronously clocked
- input latch, (D-latch), asynchronously clocked
- Flow-through input

Configuration is accomplished through the programming of EPROM architecture control bits via the logic compiler and programmer software. If synchronous operation is selected, the ILE/ICLK pin is used as a global latch/clock to all input latch/register

structures. For asynchronous operation, a separate product term in the array is used to derive the ILE/ICLK signal for each input structure. Because the clock signal for each programmable input can be individually selected, a mix between synchronously and asynchronously clocked inputs is possible. Software can configure each input structure as a flow-through input by selecting a latch and tying the ILE p-term to VCC. Data is latched/clocked on the falling edge of ILE/ICLK (synchronous mode). ILE/ICLK can function as an input to the logic array at the same time as it is used to synchronously clock the input registers.

## MACROCELLS

Each of the 24 macrocells in the device contains 8 p-terms to support logic functions and 8 p-terms for control signals. The 8 p-terms for logic functions are subdivided into 2 groups, each with 4 p-terms. This grouping of p-terms supports the proprietary p-term allocation scheme in the 5AC324. Each macrocell also provides dual feedbacks to the logic array, which results in more efficient macrocell/pin usage than possible with single feedbacks.

## Register Configuration

Each macrocell can be configured as a D, T, RS, or JK register. The 8 p-terms for control functions are organized so that 2 p-terms support each of the 4 control signals: Output Enable (OE), asynchronous I/O preset (PRESET), asynchronous I/O reset (CLEAR), and asynchronous I/O register clock (ASYNCH. CLK). Availability of 2 p-terms per control signal is another feature that increases the efficiency of the device by reducing the need to use intermediate macrocells sometimes needed to implement control functions.

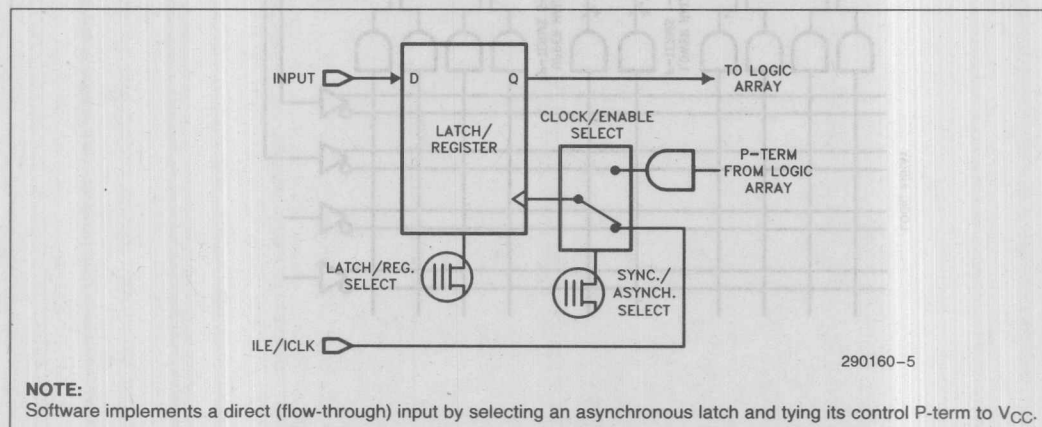


Figure 4. 5AC324 Programmable Input Structure

CLK is a global clock signal that can be used to synchronously clock any or all macrocell registers. When CLK is not used as a synchronous clock, it functions only as a dedicated input to the logic array. CLK can be used as an input to the logic array at the same time as it is used as a macrocell clock.

## Combinatorial Configuration

The macrocell register can be bypassed to implement combinatorial logic functions. When configured to provide combinatorial logic, only the OE control signal is used.

## Invert Select Bit

An invert select EPROM bit is used to invert the product term input into each macrocell register, including double inputs on JK and SR registers. This invert option allows the highest possible logic utilization by use of DeMorgan's logic inversion.

## LOGIC ARRAY

Each intersecting point in the logic array contains a programmable EPROM connection. Initially (erased state), all connections are complete, i.e., both true and complement states of all signals are connected to each p-term.

Connections are opened during programming. When both the true and complement connections exist, a logical false results on the output of the AND gate. If both the true and complement connections of a signal are programmed "open", then a logic "don't care" results for that signal. If all connections for a p-term are programmed open, then a logical true results on the output of the AND gate.

## PRODUCT TERM ALLOCATION

Product Term (p-term) allocation is defined as taking logic resources (p-terms) from macrocells where they are not used to support demand for additional p-terms in other macrocells. In the 5AC324, p-term allocation can occur in increments of 4 p-terms between adjacent macrocells. The 5AC324 includes 2 rings of 12 macrocells each. P-term groups from one macrocell can be allocated to the adjacent macrocell in the ring. P-term allocation between the two rings is not supported.

## EXAMPLE:

Figure 5 shows a p-term allocation example. In this example, the logic function in macrocell 4 requires 16 p-terms. In this case, software allocates 4 p-terms from the previous macrocell in Ring 1 (macrocell 5) and 4 p-terms from the next macrocell (macrocell 3) to accumulate a total of 16 p-terms ( $8 + 4 + 4$ ). This implementation leaves macrocells 3 and 5 with a remainder of 4 p-terms. These remaining p-terms can also be allocated away to, or supplemented with p-terms from, their adjacent macrocells in Ring 1 (macrocells 2 and 6).

With this scheme, any macrocell inside the device can support logic functions requiring between 0 and 16 p-terms. P-terms allocated away do not affect that macrocell's output structure. The input to the macrocell can be tied to VCC or GND, even when all p-terms have been allocated away. Thus the register and all control signals are still available for use if needed.

Figure 6 shows adjacent macrocells in the 5AC324. Table 1 shows the previous and next macrocells for each macrocell in the device, along with the corresponding allocation ring. P-term allocation is implemented automatically in the development software and is transparent to the user. Users can still use explicit pin assignment, but should assign pins in a way that does not conflict with p-term allocation.

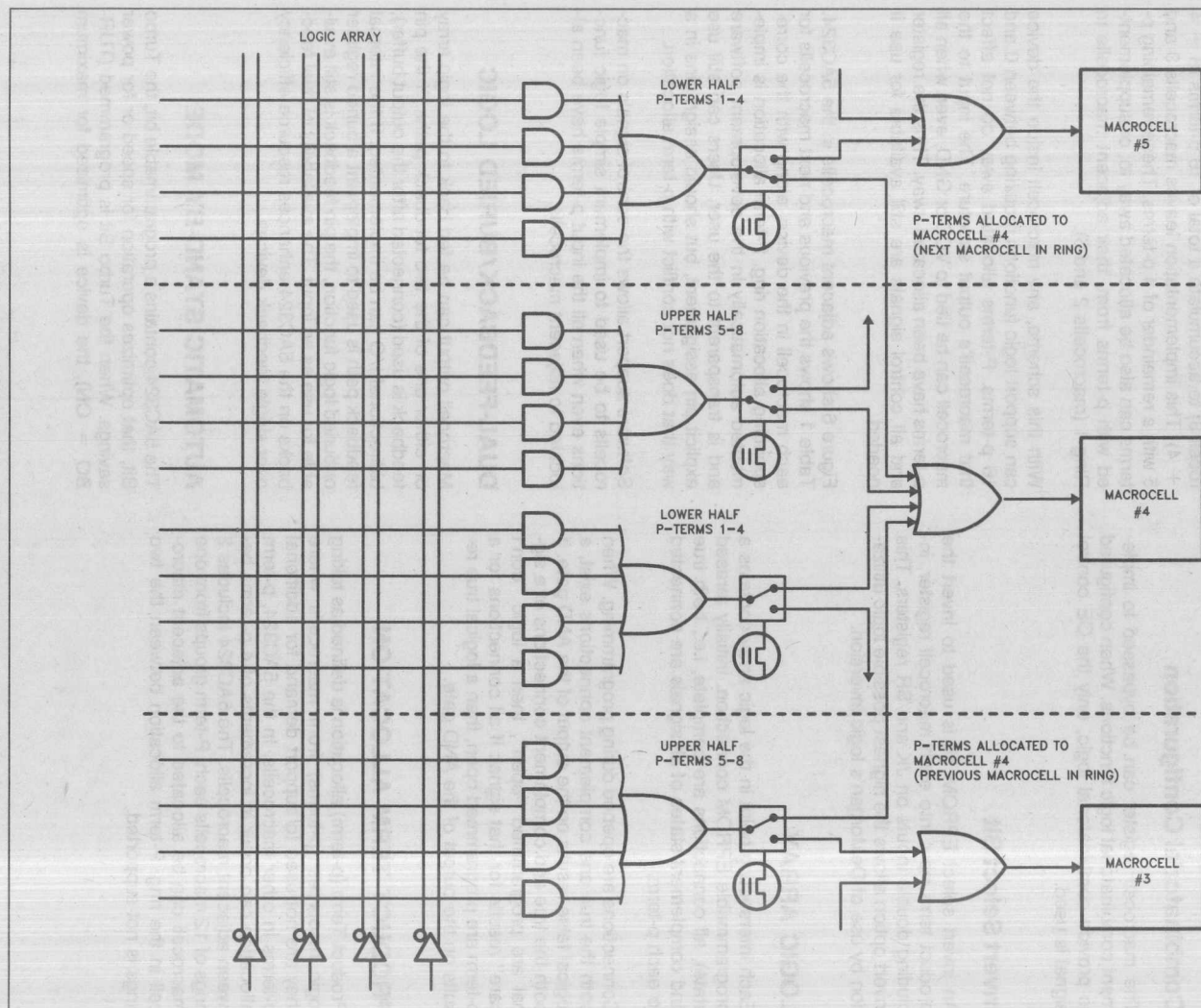
Software support allows the control signals on macrocells to be used to implement simple logic functions even when all the input p-terms have been allocated to adjacent macrocells.

## DUAL-FEEDBACK/BURIED LOGIC

Macrocell output can be fed back to the logic array on either one of the two feedback paths. If the pin feedback is used (connected after the output buffer), bidirectional I/O can be implemented. If the internal feedback path is used to implement a buried register or buried logic function, the pin feedback is still available for use as an input. The availability of dual feedbacks on the 5AC324 enhances resource efficiency over single feedback devices.

## AUTOMATIC STAND-BY MODE

The 5AC324 contains a programmable bit, the Turbo Bit, that optimizes operation for speed or for power savings. When the Turbo Bit is programmed (TURBO = ON), the device is optimized for maximum



290160-6

Figure 5. P-Term Allocation Example (8 + 4 + 4)

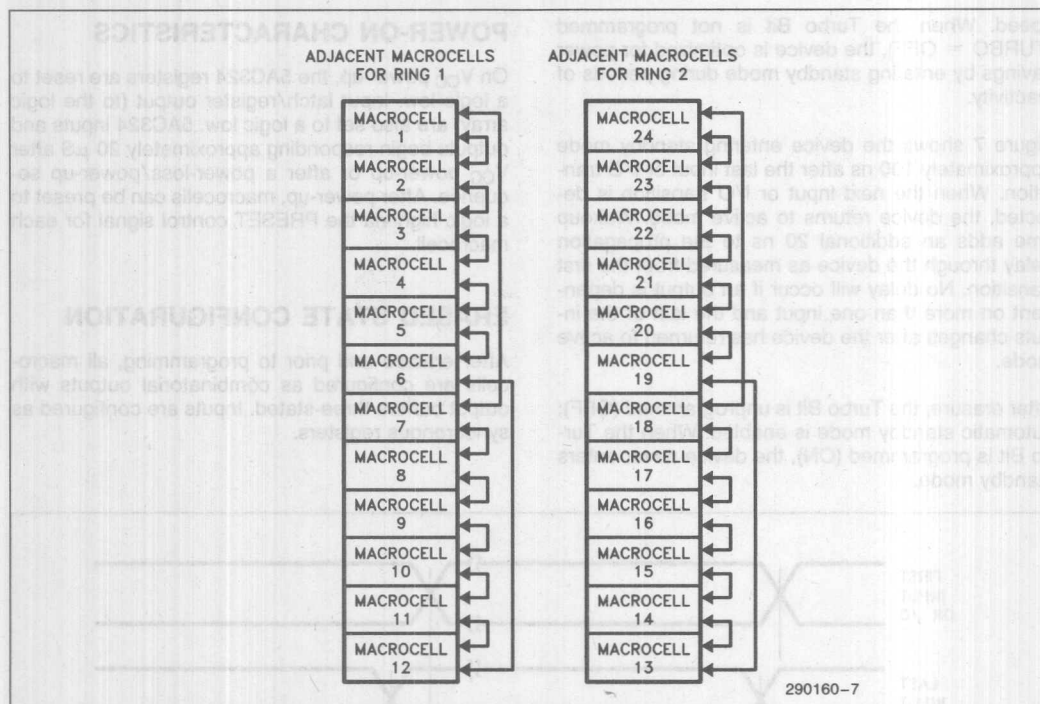


Figure 6. 5AC324 Adjacent Macrocell

Table 1. Product Term Allocation Rings

RING 1			RING 2		
Current Macrocell	Next Macrocell	Previous Macrocell	Current Macrocell	Next Macrocell	Previous Macrocell
1	7	2	13	19	14
2	1	3	14	13	15
3	2	4	15	14	16
4	3	5	16	15	17
5	4	6	17	16	18
6	5	12	18	17	24
7	8	1	19	20	13
8	9	7	20	21	19
9	10	8	21	22	20
10	11	9	22	23	21
11	12	10	23	24	22
12	6	11	24	18	23



speed. When the Turbo Bit is not programmed (TURBO = OFF), the device is optimized for power savings by entering standby mode during periods of inactivity.

Figure 7 shows the device entering standby mode approximately 100 ns after the last input or I/O transition. When the next input or I/O transition is detected, the device returns to active mode. Wakeup time adds an additional 20 ns to the propagation delay through the device as measured from the first transition. No delay will occur if an output is dependent on more than one input and the last of the inputs changes after the device has returned to active mode.

After erasure, the Turbo Bit is unprogrammed (OFF); automatic standby mode is enabled. When the Turbo Bit is programmed (ON), the device never enters standby mode.

## POWER-ON CHARACTERISTICS

On  $V_{CC}$  power-up, the 5AC324 registers are reset to a logic low. Input latch/register output (to the logic array) are also set to a logic low. 5AC324 inputs and outputs begin responding approximately 20  $\mu$ S after  $V_{CC}$  power-up or after a power-loss/power-up sequence. After power-up, macrocells can be preset to a logic high via the PRESET control signal for each macrocell.

## ERASED STATE CONFIGURATION

After erasure and prior to programming, all macrocells are configured as combinatorial outputs with output buffers three-stated. Inputs are configured as synchronous registers.

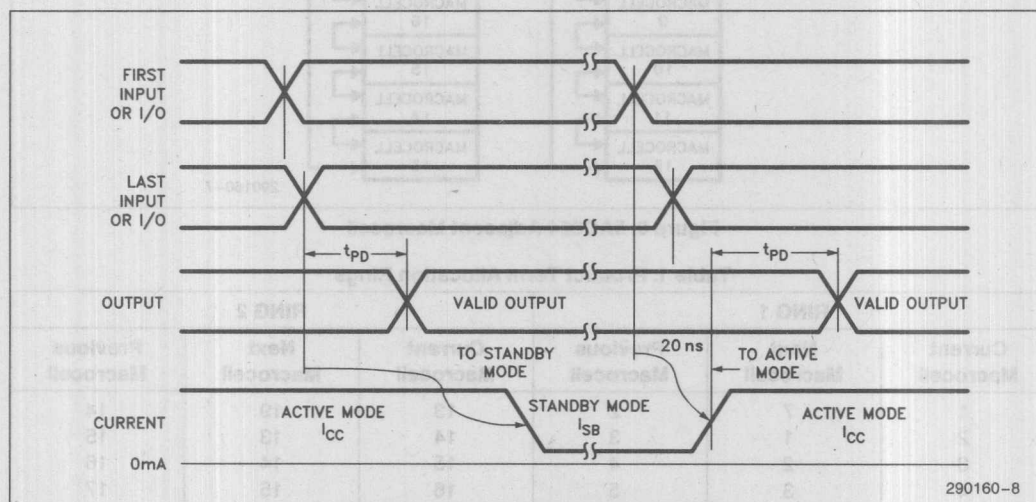


Figure 7. 5AC324 Standby and Active Mode Transitions

## ERASURE CHARACTERISTICS

Erasure time for the 5AC324 is 1 hour at 12,000  $\mu\text{W}/\text{cm}^2$  with a 2537Å UV lamp.

Erasure characteristics of the device are such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000Å. It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000Å-4000Å range. Data shows that constant exposure to room level fluorescent lighting could erase the typical 5AC324 in approximately six years, while it would take approximately two weeks to erase the device when exposed to direct sunlight. If the device is to be exposed to these lighting conditions for extended periods of time, conductive opaque labels should be placed over the device window to prevent unintentional erasure.

The recommended erasure procedure for the 5AC324 is exposure to shortwave ultraviolet light with a wavelength of 2537Å. The integrated dose (i.e., UV intensity x exposure time) for erasure should be a minimum of forty (40) Wsec/cm<sup>2</sup>. The erasure time with this dosage is approximately 1 hour using an ultraviolet lamp with a 12,000  $\mu\text{W}/\text{cm}^2$  power rating. The device should be placed within 1 inch of the lamp tubes during exposure. The maximum integrated dose the 5AC324 can be exposed to without damage is 7258 Wsec/cm<sup>2</sup> (1 week at 12,000  $\mu\text{W}/\text{cm}^2). Exposure to high intensity UV light for longer periods may cause permanent damage to the device.$

## Intelligent Programming Algorithm

The 5AC324 supports the Intelligent Programming Algorithm, which rapidly programs Intel EPLDs, while maintaining a high degree of reliability. It is particularly suited for production programming environments. This method ensures reliability as the incremental programming margin of each bit has been verified during programming. Programming voltage and waveform specifications are available by request from Intel to support programming the device.

## LATCH-UP IMMUNITY

All of the input, I/O, and clock pins of the device have been designed to resist latch-up which is inher-

ent in inferior CMOS structures. The 5AC324 is designed with Intel's proprietary 1-micron CHMOS EPROM process. Thus, each of the pins will not experience latch-up with currents up to  $\pm 100$  mA and voltages ranging from  $-0.5\text{V}$  to  $(V_{CC} + 0.5\text{V})$ . The programming pin is designed to resist latch-up to the 13.5V maximum device limit.

## DESIGN RECOMMENDATIONS

For proper operation, it is recommended that all input and output pins be constrained to the voltage range  $\text{GND} \leq (V_{IN} \text{ or } V_{OUT}) \leq V_{CC}$ . All unused inputs and I/Os should be tied high or low to minimize power consumption (do not leave them floating). A power supply decoupling capacitor of at least 0.2 $\mu\text{F}$  must be connected directly between each  $V_{CC}$  and GND pin.

As with all CMOS devices, ESD handling procedures should be used with the 5AC324 to prevent damage to the device during programming, assembly, and test.

## FUNCTIONAL TESTING

Since the logical operation of the 5AC324 is controlled by EPROM elements, the device is completely testable during the manufacturing process. Each programmable EPROM bit controlling the internal logic is tested using application independent test patterns. EPROM cells in the device are 100% tested for programming and erasure. After testing, the devices are erased before shipments to the customers. No post-programming tests of the EPROM array are required.

The testability and reliability of EPROM-based programmable logic devices is an important feature over similar devices based on fuse technology. Fuse-based programmable logic devices require a user to perform post-programming tests to insure device functionality. During the manufacturing process, tests on fuse-based parts can only be performed in very restricted ways in order to avoid pre-programming the array.

## ADF PRIMITIVES SUPPORTED

The following ADF primitives are supported by this device:

INP	NOTF
LINP	JOJF
RINP	JONF
CONF	SONF
COCF	SOSF
COIF	TOIF
RONF	TONF
ROIF	TOTF
RORF	CLKB
NOCF	LINB
NORF	
NOJF	
NOSF	

## SOFTWARE SUPPORT

Full logic compilation and functional simulation for the 5AC324 is supported by PLDshell Plus™ software. The GUPI 40D44J provides programming support on Intel programmers.

PLDshell Plus design software is Intel's new, user-friendly design tool for  $\mu$ PLD design. PLDshell Plus allows users to incorporate their preferred text editor, programming software, and additional design tools into an easy-to-use, menued design environment that includes Intel's PLDasm™ logic compiler and simulation software along with disassembly, conversion, and translation utilities. The PLDasm compiler and simulator software accepts industry-standard PDS source files that express designs as Boolean equations, truth tables, or state machines. On-line help, datasheet briefs, technical notes, and error message information, along with waveform viewing/printing capability make the design task as easy as possible. PLDshell Plus software is available from Intel Literature channels or from your local Intel sales representative.

Tools that support schematic capture and timing simulation for the 5AC324 are available. Support under iPLS II is still available. Please refer to the "Development Tools" section of the Programmable Logic handbook.

The 5AC324 is also supported by third-party logic compilers such as ABEL†, CUPL†, PLDesigner†, Log/IC, etc. Programming support is provided by third-party programmer companies such as Data I/O, Logical Devices, STAG, etc. Please refer to the "Third-Party Support" lists in the *Programmable Logic* handbook for complete information and vendor contacts.

## ORDERING INFORMATION

t <sub>PD</sub> (ns)	t <sub>CO</sub> (ns)	f <sub>MAX</sub> (MHz)	Order Code	Package	Operating Range
25	17.8	66	N5AC324-25	PLCC	Commercial
			P5AC324-25	PDIP	
			D5AC324-25	*CERDIP	
30	20	50	N5AC324-30	PLCC	Commercial
			P5AC324-30	PDIP	
			D5AC324-30	*CERDIP	

\*Windowed package allows UV erase.

\*PLDshell Plus™ is a trademark of Intel Corporation.

†ABEL is a trademark of Data I/O Corporation. CUPL is a trademark of Logical Devices, Inc. PLDesigner is a trademark of MINC, Inc. Log/IC is a trademark of ISDATA, Inc.

**ABSOLUTE MAXIMUM RATINGS\***

Supply Voltage ( $V_{CC}$ )<sup>(1)</sup> ..... -2.0V to +7.0V  
 Programming Supply  
 Voltage ( $V_{PP}$ )<sup>(1)</sup> ..... -2.0V to +13.5V  
 D.C. Input Voltage ( $V_I$ )<sup>(1,2)</sup> .... -0.5V to  $V_{CC} + 0.5V$   
 Storage Temperature ( $T_{stg}$ ) ..... -65°C to +150°C  
 Ambient Temperature ( $T_{amb}$ )<sup>(3)</sup> ... -10°C to +85°C

**NOTES:**

1. Voltage with respect to GND.
2. Minimum D.C. input is -0.5V. During transitions, the inputs may undershoot to -2.0V or overshoot to +7.0V less than 20 ns under no load conditions.
3. Under bias. Extended Temperature versions are also available.

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

*\*WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

**RECOMMENDED OPERATING CONDITIONS**

Symbol	Parameter	Min	Max	Unit
$V_{CC}$	Supply Voltage	4.75	5.25	V
$V_{IN}$	Input Voltage	0	$V_{CC}$	V
$V_O$	Output Voltage	0	$V_{CC}$	V
$T_A$	Operating Temperature	0	+70	°C
$t_R$	Input Rise Time		500	ns
$t_F$	Input Fall Time		500	ns

**PACKAGE/TECHNOLOGY SPECIFICATIONS**

Description	Specification
$\theta_{JA}$ —Junction-to-Ambient Thermal Resistance	34°C/W—CerDIP 43°C/W—PDIP 43°C/W—PLCC
$\theta_{JC}$ —Junction-to-Case Thermal Resistance	13°C/W—CerDIP 15°C/W—PDIP 15°C/W—PLCC
$I_{CC}$ Hot—Ambient @70°C	150 mA
$I_{CC}$ Typical—Ambient @25°C	150 mA
Process	CHMOS IIIIE, PX 29

**D.C. CHARACTERISTICS** ( $T_A = 0^\circ\text{C}$  to  $+70^\circ\text{C}$ ,  $V_{CC} = 5.0V \pm 5\%$ )

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
$V_{IH}^{(4)}$	High Level Input Voltage	2.0		$V_{CC} + 0.3$	V	
$V_{IL}^{(4)}$	Low Level Input Voltage	-0.3		0.8		
$V_{OH}^{(5)}$	High Level Output Voltage	2.4			V	$I_O = -4.0$ mA D.C., $V_{CC} = \text{min.}$
$V_{OL}$	Low Level Output Voltage			0.45	V	$I_O = 8.0$ mA D.C., $V_{CC} = \text{min.}$
$I_I$	Input Leakage Current			$\pm 10$	$\mu\text{A}$	$V_{CC} = \text{max.},$ $\text{GND} < V_{IN} < V_{CC}$
$I_{OZ}$	Output Leakage Current			$\pm 10$	$\mu\text{A}$	$V_{CC} = \text{max.},$ $\text{GND} < V_{OUT} < V_{CC}$
$I_{SC}^{(6)}$	Output Short Circuit Current	-30		-90	mA	$V_{CC} = \text{max.}, V_{OUT} = 0.5V$



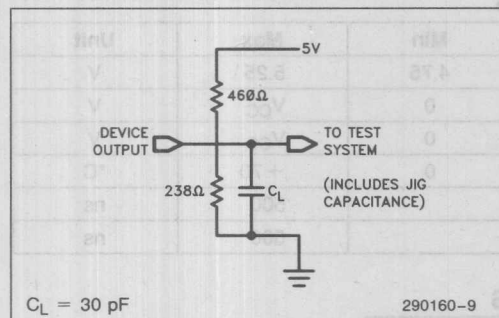
# D.C. CHARACTERISTICS (T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5.0V ± 5%) (Continued)

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
I <sub>SB</sub> <sup>(7)</sup>	Standby Current		150	500	μA	V <sub>CC</sub> = max., V <sub>IN</sub> = V <sub>CC</sub> or GND, Standby Mode
I <sub>CC</sub>	Power Supply Current (See I <sub>CC</sub> vs Freq. Graph)		20		mA	V <sub>CC</sub> = max., V <sub>IN</sub> = V <sub>CC</sub> or GND, No Load, f <sub>IN</sub> = 1 MHz, Active Mode (Turbo = Off), Device Prog. as Two 12-Bit Counters

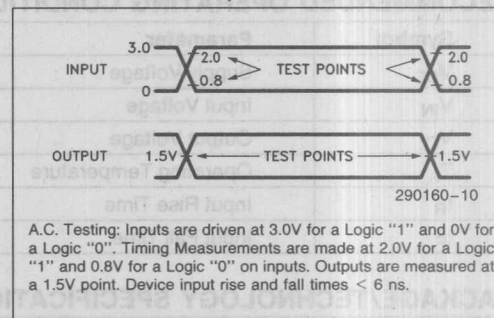
## NOTES:

4. Absolute values with respect to device GND; all over and undershoots due to system or tester noise are included.
5. I<sub>O</sub> at CMOS levels (3.84V) = -2 mA.
6. Not more than 1 output should be tested at a time. Duration of that test should not exceed 1 second.
7. With Turbo Bit Off, device automatically enters standby mode approximately 100 ns after last input transition.

## A.C. TESTING LOAD CIRCUIT



## A.C. TESTING INPUT, OUTPUT WAVEFORM



## CAPACITANCE

Symbol	Parameter	Min	Typ	Max	Unit	Conditions
C <sub>IN</sub>	Input Capacitance			8	pF	V <sub>IN</sub> = 0V, f = 1.0 MHz
C <sub>OUT</sub>	I/O Capacitance			15	pF	V <sub>OUT</sub> = 0V, f = 1.0 MHz
C <sub>CLK</sub>	Clock Pin Capacitance			15	pF	V <sub>IN</sub> = 0V, f = 1.0 MHz
C <sub>VPP</sub>	V <sub>PP</sub> Pin (LIN3)			25	pF	V <sub>IN</sub> = 0V, f = 1.0 MHz

## COMBINATORIAL MODE A.C. CHARACTERISTICS

(T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5.0V ± 5%, Turbo Bit On)<sup>(8)</sup>

Symbol	Parameter	5AC324-25			5AC324-30			5AC324-35			Non-Turbo <sup>(9)</sup> Mode	Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
t <sub>PD</sub>	Input or I/O to Output Valid		20	25		25	30		30	35	+ 20	ns
t <sub>PZX</sub> <sup>(10)</sup>	Input or I/O to Output Enable		20	25		25	30		30	35	+ 20	ns
t <sub>PXZ</sub> <sup>(10)</sup>	Input or I/O to Output Disable		20	25		25	30		30	35	+ 20	ns
t <sub>CLR</sub>	Asynch. Reset to Q Reset		20	25		25	30		30	35	+ 20	ns
t <sub>SET</sub>	Asynch. Set to Q Set		20	25		25	30		30	35	+ 20	ns

## NOTES:

8. Typical values are at T<sub>A</sub> = +25°C, V<sub>CC</sub> = 5V, Active Mode.
9. If device is operated with Turbo bit Off (Non-Turbo Mode) and the device is inactive for approx. 100 ns, increase time by amount shown.
10. t<sub>PZX</sub> and t<sub>PXZ</sub> measured at ±0.5V from steady-state voltage as driven by spec. output load. t<sub>PXZ</sub> measured with C<sub>L</sub> = 5 pF.

**SYNCHRONOUS CLOCK MODE (MACROCELLS) A.C. CHARACTERISTICS**(T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5.0V ± 5%, Turbo Bit On)<sup>(8)</sup>

Symbol	Parameter	5AC324-25			5AC324-30			5AC324-35			Non-Turbo <sup>(9)</sup> Mode	Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
f <sub>MAX</sub>	Maximum Frequency (Pipelined) (1/t <sub>CW</sub> )—No Feedback		80	66		66	50		50	40		MHz
f <sub>CNT1</sub>	Maximum Frequency (1/t <sub>SU</sub> + t <sub>CO</sub> ) —External Feedback		40	33		33.3	25		27	21.2		MHz
f <sub>CNT2</sub>	Maximum Frequency (1/t <sub>CNT</sub> ) —Internal Feedback		40	33.3		33.3	28.5		28.5	25		MHz
t <sub>SU1</sub>	Input Setup Time to CLK ↑	12.5	10		20	15		25	20		+ 20	ns
t <sub>SU2</sub>	I/O Setup Time to CLK ↑	12	10		20	15		25	20		+ 20	ns
t <sub>H</sub>	Input or I/O Hold Time from CLK ↑	0			0			0				ns
t <sub>CO</sub>	CLK ↑ to Output Valid		15	17.8		15	20		17	22		ns
t <sub>CNT</sub>	Register Output Feedback to Register Input— Internal Path		25	30		30	35		35	40	+ 20	ns
t <sub>CH</sub>	Clock High Time	7			9			11				ns
t <sub>CL</sub>	Clock Low Time	7			9			11				ns
t <sub>CW</sub>	Minimum Clock Period	15			20			25				ns

2

**SYNCHRONOUS CLOCK MODE (INPUT STRUCTURE) A.C. CHARACTERISTICS**(T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5.0V ± 5%, Turbo Bit On)<sup>(8)</sup>

Symbol	Parameter	5AC324-25			5AC324-30			5AC324-35			Non-Turbo <sup>(9)</sup> Mode	Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
f <sub>MAXI</sub>	Maximum Frequency (1/t <sub>CWI</sub> )		80	66		60	50		50	40		MHz
t <sub>SUIR</sub>	Input Register Setup Time Before ILE/ICLK ↓	1			2.5			5				ns
t <sub>ESUI</sub> <sup>(11)</sup>	Input Latch Setup Time Before ILE ↑	1			2.5			5				ns
t <sub>COI</sub>	ICLK ↓ to Comb. Output		25	30		30	35		35	40	+ 20	ns
t <sub>EOI</sub>	ILE ↑ to Comb. Output		25	30		30	35		35	40	+ 20	ns
t <sub>HI</sub>	Input Hold after ICLK ↓	8			9			10				ns
t <sub>EHI</sub>	Input Hold after ILE ↓	7			8			9				ns
t <sub>CHI</sub>	ILE/ICLK High Time	7			9			11				ns
t <sub>CLI</sub>	ILE/ICLK Low Time	7			9			11				ns
t <sub>CWI</sub>	Minimum Input Clock Period	15			20			25				ns

**ASYNCHRONOUS CLOCK MODE (MACROCELLS) A.C. CHARACTERISTICS**(T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5.0V ±5%, Turbo Bit On)<sup>(8)</sup>

Symbol	Parameter	5AC324-25			5AC324-30			5AC324-35			Non-Turbo <sup>(9)</sup> Mode	Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
f <sub>AMAX</sub>	Max. Frequency (Pipelined) (1/t <sub>ACW</sub> )—No Feedback		80	66		60	50		50	40		MHz
f <sub>ACNT1</sub>	Max. Frequency (1/t <sub>ASU</sub> + t <sub>ACO</sub> ) External Feedback		32.2	27.7		27	23.8		22.2	20		MHz
f <sub>ACNT2</sub>	Max. Frequency (1/t <sub>ACNT</sub> ) Internal Feedback		40	33.3		33.3	28.5		28.5	25		MHz
t <sub>ASU1</sub>	Input Setup Time to Asynch. CLK	11			12			15			+ 20	ns
t <sub>ASU2</sub>	I/O Setup Time to Asynch. CLK	11			12			15			+ 20	ns
t <sub>AH</sub>	Input or I/O Hold Time from Asynch. CLK	3	0		4	0		5	0			ns
t <sub>ACO</sub>	Asynch. CLK to Output Valid		20	25		25	30		30	35	+ 20	ns
t <sub>ACNT</sub>	Asynch. Output Feedback to Register Input - Internal Path		25	30		30	35		35	40	+ 20	ns
t <sub>ACH</sub>	Asynch. CLK High Time	7			9			11			+ 20	ns
t <sub>ACL</sub>	Asynch. CLK Low Time	7			9			11			+ 20	ns
t <sub>ACW</sub>	Asynch. CLK Period	15			20			25			+ 20	ns

**ASYNCHRONOUS CLOCK MODE (INPUT STRUCTURE) A.C. CHARACTERISTICS**(T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5.0V ±5%, Turbo Bit On)<sup>(8)</sup>

Symbol	Parameter	5AC324-25			5AC324-30			5AC324-35			Non-Turbo <sup>(9)</sup> Mode	Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
f <sub>AMAXI</sub>	Maximum Frequency Input Register (1/t <sub>ACWI</sub> )		80	66		60	50		50	40		MHz
t <sub>ASUIR</sub>	Input Register Setup Time Before Asynch. ICLK	-5			-5			-5				ns
t <sub>AESUI</sub> <sup>(11)</sup>	Input Latch Setup Time Before Asynch. ILE	-5			-5			-5				ns
t <sub>ACOI</sub>	Asynch. ICLK to Comb. Output		25	30		30	35		45	50	+ 20	ns
t <sub>AEOI</sub>	Asynch. ILE to Comb. Output		25	30		30	45		45	50	+ 20	ns
t <sub>AHI</sub>	Input Hold after Asynch. ICLK	15			18			20				ns
t <sub>AHEI</sub>	Input Hold after Asynch. ILE	14			17			19				ns
t <sub>ACHI</sub>	Asynch. ILE/ICLK High Time	7			9			11			+ 20	ns
t <sub>ACLI</sub>	Asynch. ILE/ICLK Low Time	7			9			11			+ 20	ns
t <sub>ACWI</sub>	Minimum Input Clock Period	15			20			25			+ 20	ns

**NOTE:**11. This specification must be met to guarantee t<sub>EOI</sub>. When ILE goes high before data is valid, use t<sub>PD</sub> instead of t<sub>EOI</sub>.

# **INPUT-CLOCK-TO-MACROCELL-CLOCK A.C. CHARACTERISTICS**

( $T_A = 0^\circ\text{C}$  to  $+70^\circ\text{C}$ ,  $V_{CC} = 5.0\text{V} \pm 5\%$ , Turbo Bit On)<sup>(8)</sup>

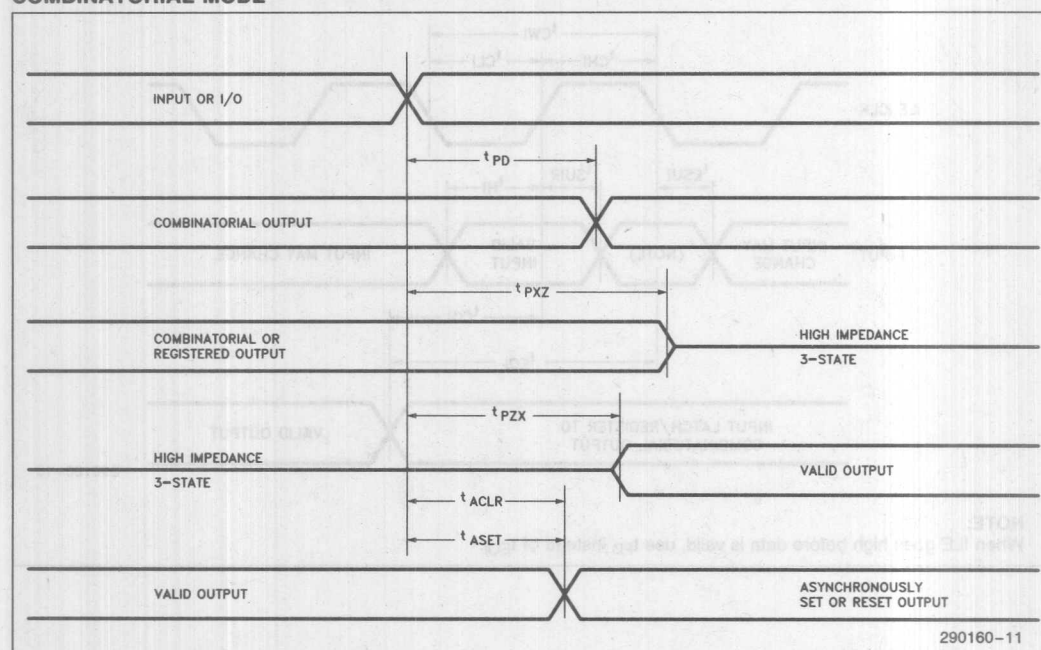
Symbol	Parameter	5AC324-25			5AC324-30			5AC324-35			Non-Turbo <sup>(9)</sup> Mode	Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
$t_{C1C2}^{(12)}$	Synchronous ILE/ICLK to Synchronous Macrocell CLK	20			25			30			+ 20	ns
	Synchronous ILE/ICLK to Asynchronous Macrocell CLK	12.5			15			18			+ 20	ns
	Asynchronous ILE/ICLK to Synchronous Macrocell CLK	40			45			50			+ 20	ns
	Asynchronous ILE/CLK to Asynchronous Macrocell CLK	20			25			30			+ 20	ns

## **NOTE:**

12. Times for SETUP, HOLD, and OUTPUT VALID are shown in previous tables.

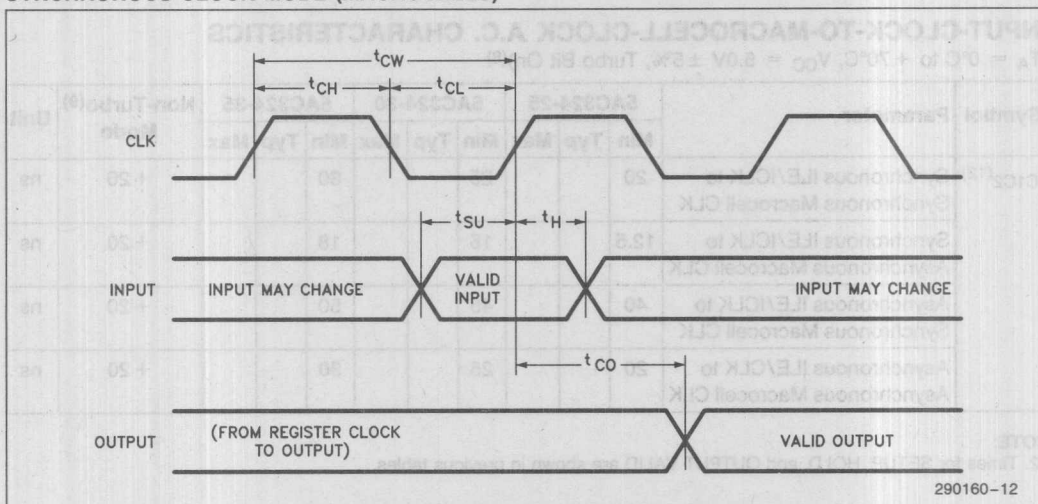
## **SWITCHING WAVEFORMS**

### **COMBINATORIAL MODE**

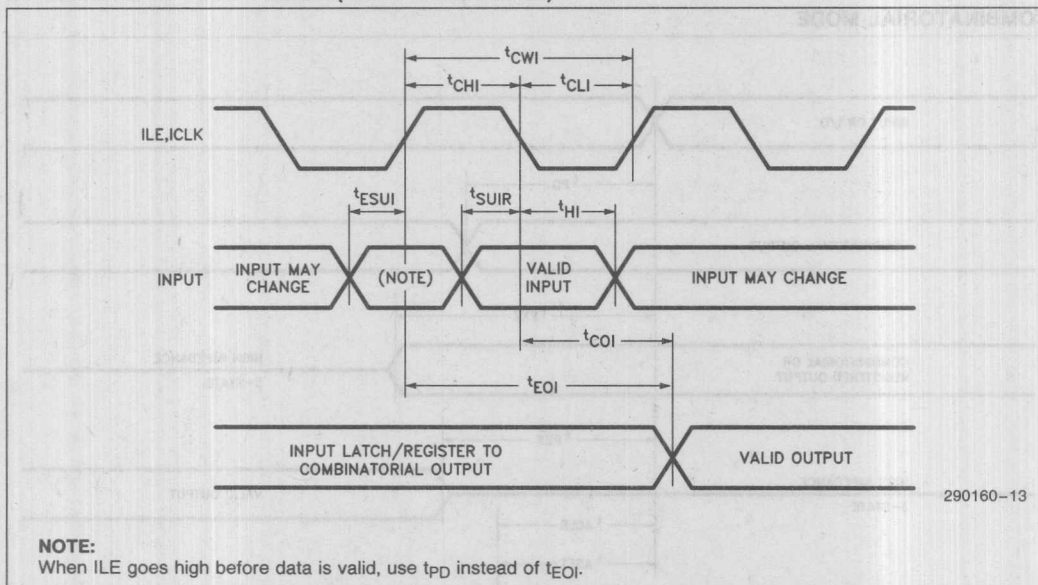




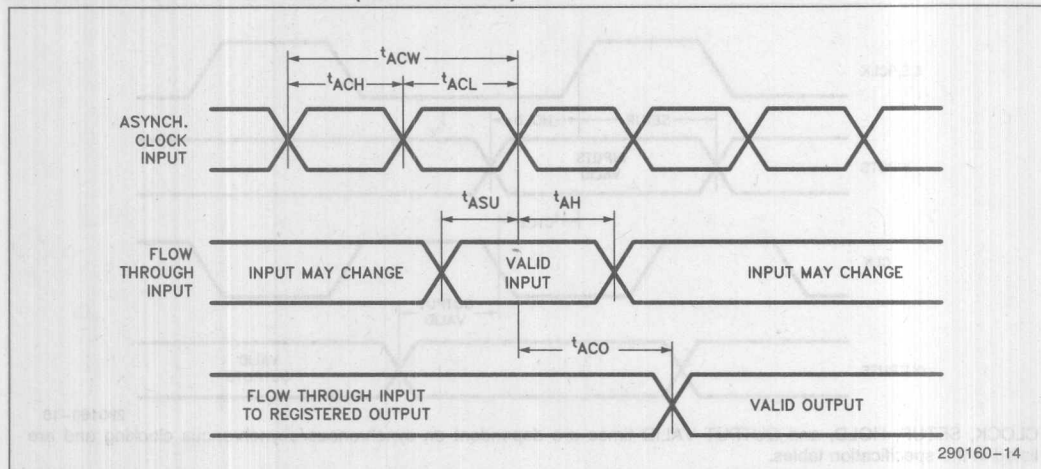
## SYNCHRONOUS CLOCK MODE (MACROCELLS)



## SYNCHRONOUS CLOCK MODE (INPUT STRUCTURE)

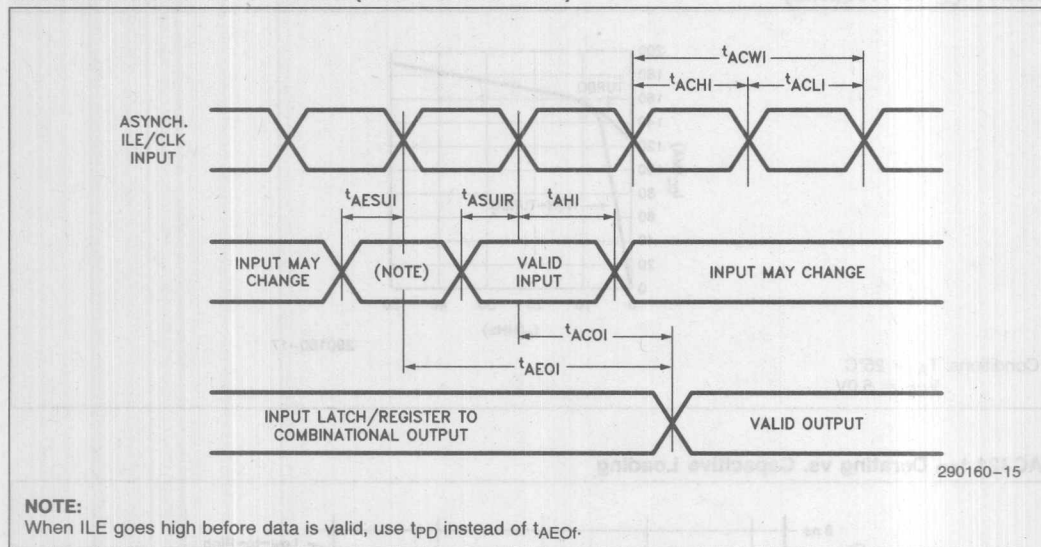


# ASYNCHRONOUS CLOCK MODE (MACROCELLS)

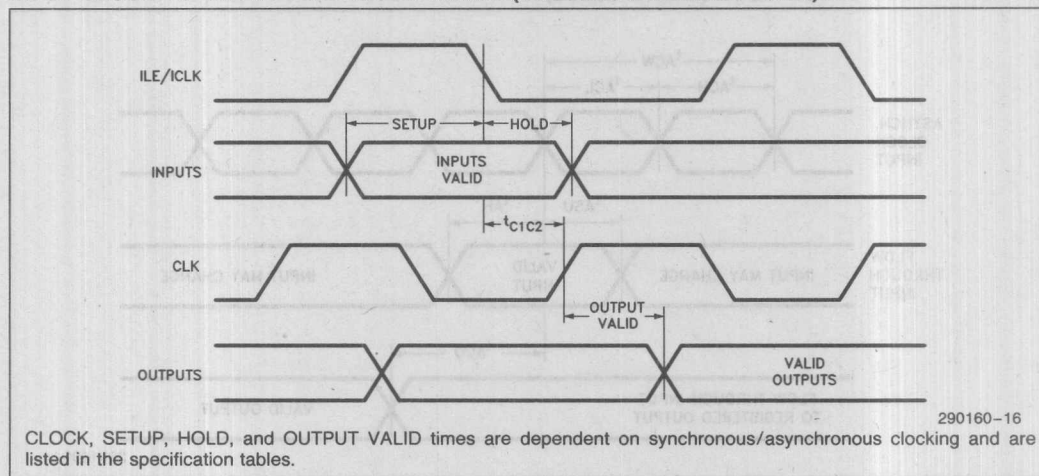


2

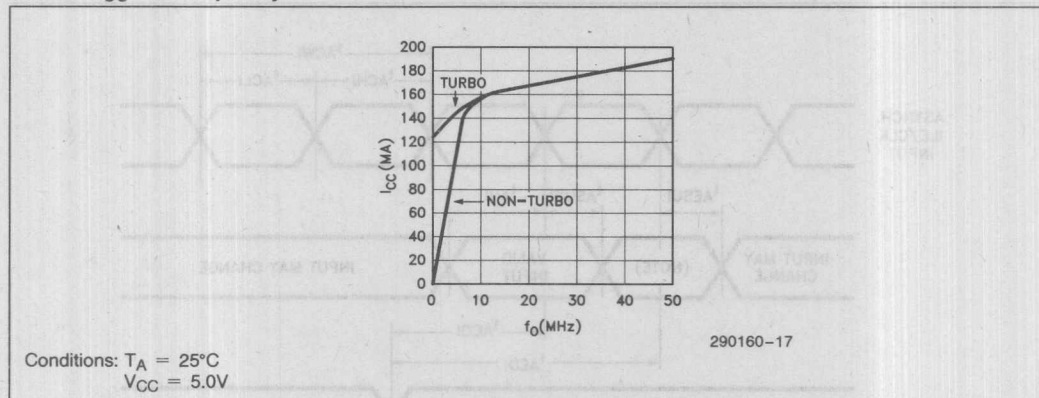
# ASYNCHRONOUS CLOCK MODE (INPUT STRUCTURE)



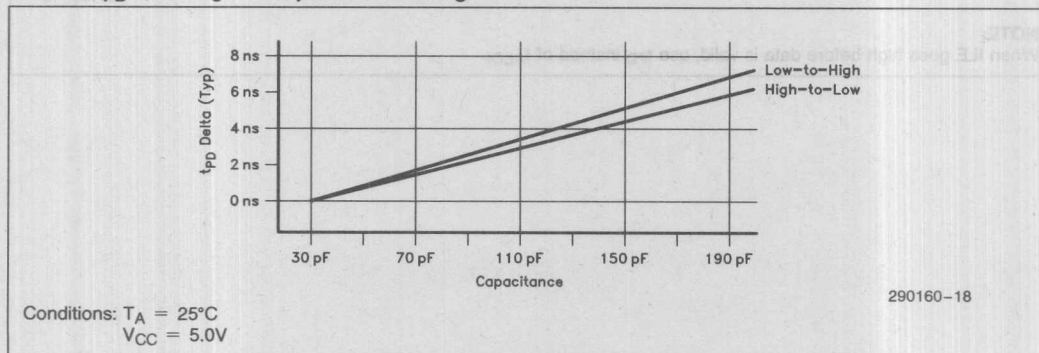
# INPUT-CLOCK-TO-MACROCELL CLOCK TIMING (CLOCKED PIPELINED DATA)



## 5AC324 $I_{CC}$ vs. Frequency



## 5AC324 $t_{PD}$ Derating vs. Capacitive Loading



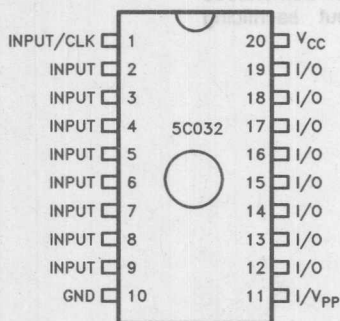
# 5C032

## 8-MACROCELL CHMOS EPLD

- High-Density, Low-Power Replacement for SSI & MSI Devices and Bipolar PLDs
- Up to 18 Inputs (10 Dedicated & 8 I/O) and 8 Outputs
- Eight Macrocells with Programmable I/O Architecture
- $t_{PD} = 30$  ns (max), 43.5 MHz Pipelined, 28.5 MHz with Feedback
- Low Power Upgrade for All Commonly Used 20-pin PLDs
- CHMOS EPROM Technology Based UV Erasable (CerDIP)

- Programmable "Security Bit" Allows Total Protection of Proprietary Designs
- $I_{CC}$  (standby) 100  $\mu$ A (max)  
 $I_{CC}$  (10 MHz) 25 mA (max)
- 100% Generically Tested EPROM Logic Control Array
- 20-pin 0.3" Ceramic and Plastic DIP Package  
(See Packaging Spec., Order #240800)  
Package Type D and P
- 100% Compatible with EP320

2



Pin Configuration

290155-1



The Intel 5C032 is an 8-macrocell, 20-pin, general-purpose EPLD (Erasable Programmable Logic Device). This device can be used to replace bipolar programmable logic arrays and LS TTL and 74HC (CMOS) SSI and MSI logic devices. The 5C032 can also be used as a direct, low-power replacement for almost all common 20-pin fuse-based programmable logic devices. With its flexible programmable I/O architecture, this device is a superset of common 20-pin PLDs.

The 5C032 EPLD uses CHMOS EPROM (floating gate) cells as logic control elements instead of fuses. The CHMOS EPROM technology reduces power consumption of EPLDs to less than 20% of a comparable bipolar device without sacrificing speed performance. In addition, the use of Intel's advanced CHMOS II-E EPROM process technology enables greater logic densities to be achieved with superior speed and low-power performance over other comparable devices. Intel's 5C032 has the benefit of "zero" stand-by power not available on other programmable logic devices. EPROM technology allows these devices to be 100% factory tested by programming and erasing all the EPROM logic control elements.

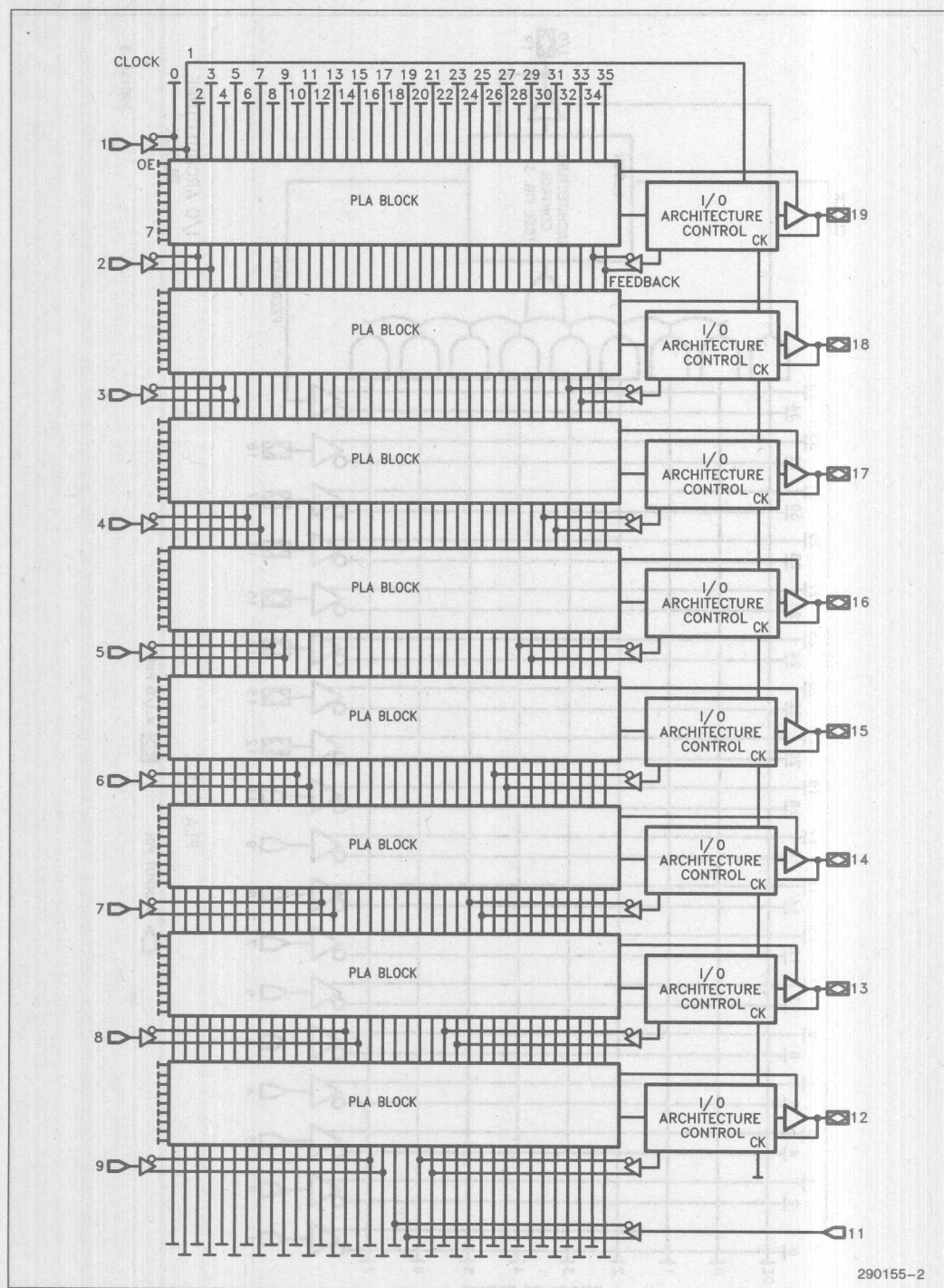
The 5C032 with its superior speed and power performance and its plastic package is an ideal production vehicle for high-volume manufacturing. Most commonly used 20-pin bipolar PLDs can be easily replaced with this device allowing for tremendous power consumption savings without sacrificing speed of operation.

## ARCHITECTURE DESCRIPTION

The architecture of the 5C032 is based on the "Sum of Products" PLA (Programmable Logic Array) structure with a programmable AND array feeding into a fixed OR array. This device can accommodate both combinational and sequential logic functions. A proprietary programmable I/O architecture provides individual selection of either combinational or registered output and feedback signals, all with selectable polarity.

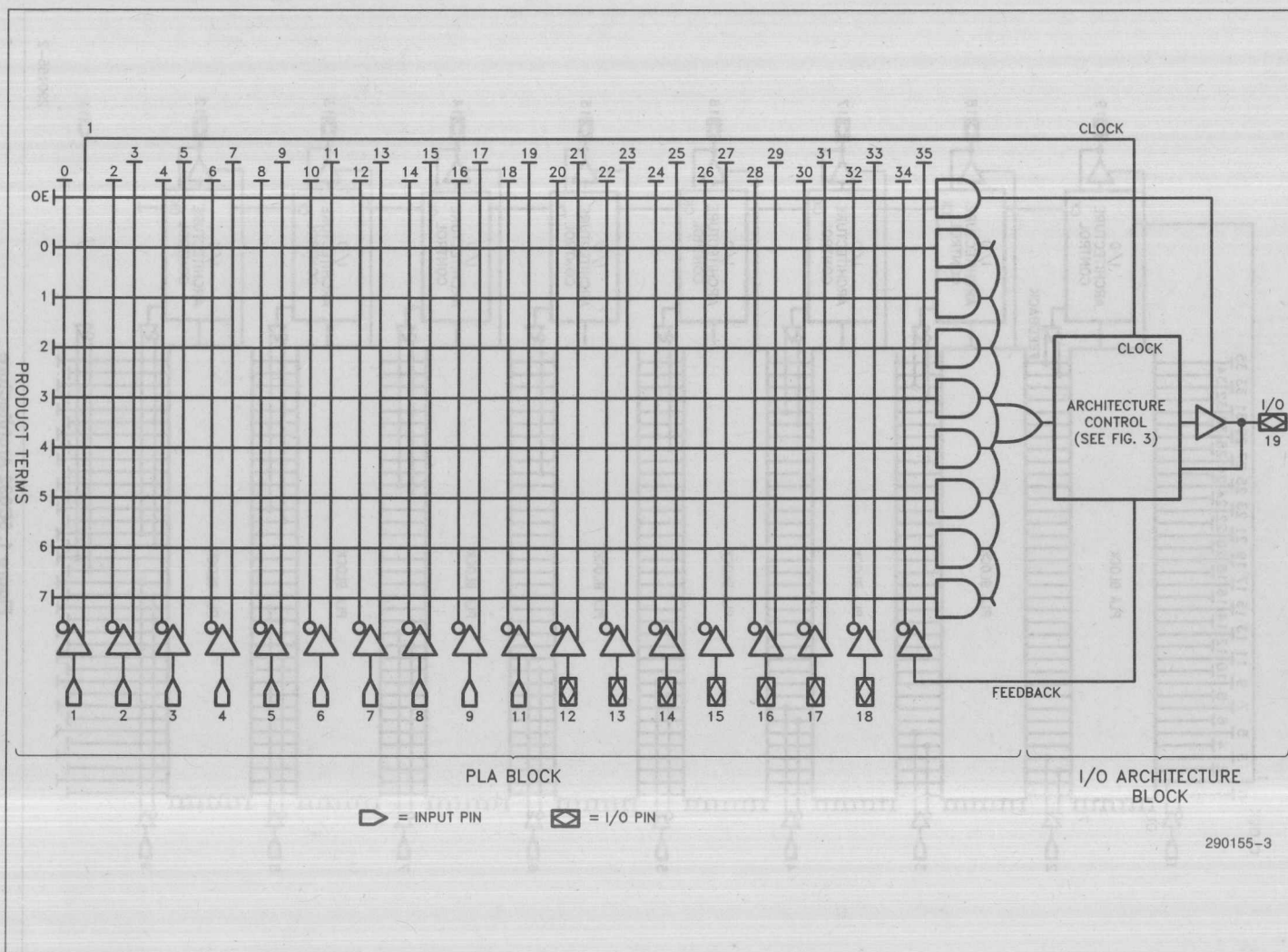
The 5C032 contains 10 dedicated inputs as well as 8 input/output pins. These I/O pins can be individually configured to be inputs, outputs or bi-directional I/O pins. Each of these I/O pins is connected to a macrocell. The 5C032 contains 8 identical macrocells organized as shown in Figure 1.

Each macrocell (see Figure 2) consists of a PLA (programmable logic array) block and an I/O architecture block, which contains a "D" type register. The PLA block consists of eight 36-input AND gates (TRUE & COMPLEMENT of 10 dedicated inputs plus the 8 feedback inputs from the eight macrocells), feeding into an OR gate. The output of this PLA block is fed into the I/O architecture block. The different I/O and feedback options that are available in the 5C032 I/O block are shown in Figure 3.



290155-2

Figure 1. 5C032 Architecture



290155-3

Figure 2. Logic Array Macrocell

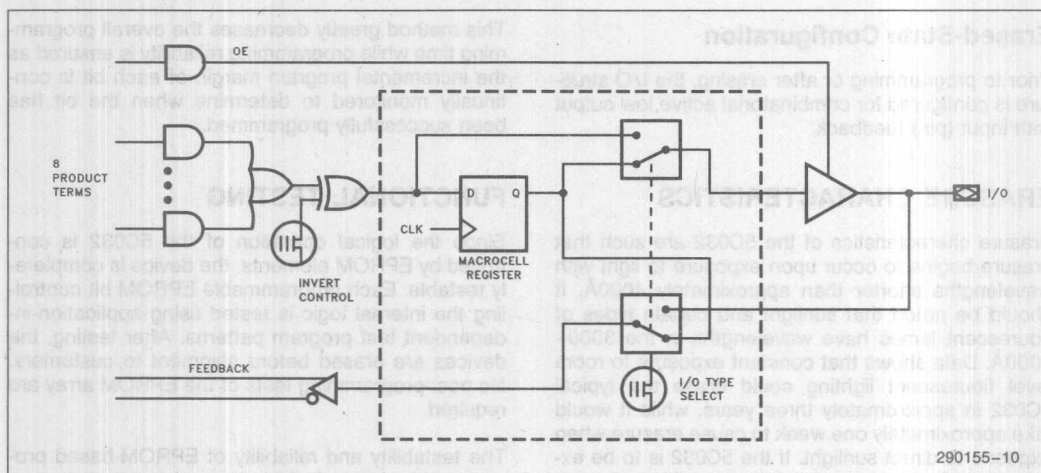


Figure 3. 5C032 I/O Architecture Control

## 20 PIN CMOS COMPATIBILITY

The 5C032 is architected to be a logical superset of most 20 pin bipolar programmable array logic (PAL\*) devices. The I/O and logic sections of the 5C032 device can be configured to emulate any of the devices listed below. Designers can make use of this feature by reducing the power of PAL based systems (EPLDs are much lower power), replacing multiple PAL inventory items with a single EPLD. Designers can also create new 20 pin PLD configurations by utilizing the individual logic and output controls of each macrocell.

List of PAL devices logically compatible with the 5C032.

16V8	16L2
10H8	16L8
12H6	16R8
14H4	16R6
16H2	16R4
16H8	16P8A
16C1	16RP8A
10LB	16RP6A
12L6	16RP4A
14L4	

\*PAL is a registered trademark of Advanced Micro Devices.



## Erased-State Configuration

Prior to programming or after erasing, the I/O structure is configured for combinatorial active low output with input (pin) feedback.

## ERASURE CHARACTERISTICS

Erasure characteristics of the 5C032 are such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000Å. It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000–4000Å. Data shows that constant exposure to room level fluorescent lighting could erase the typical 5C032 in approximately three years, while it would take approximately one week to cause erasure when exposed to direct sunlight. If the 5C032 is to be exposed to these types of lighting conditions for extended periods of time, conductive opaque labels should be placed over the device window to prevent unintentional erasure.

The recommended erasure procedure for the 5C032 is exposure to shortwave ultraviolet light with a wavelength of 2537Å. The integrated dose (i.e., UV intensity  $\times$  exposure time) for erasure should be a minimum of fifteen (15) Wsec/cm<sup>2</sup>. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12,000  $\mu$ W/cm<sup>2</sup> power rating. The 5C032 should be placed within one inch of the lamp tubes during erasure. The maximum integrated dose the 5C032 can be exposed to without damage is 7258 Wsec/cm<sup>2</sup> (1 week at 12,000  $\mu$ W/cm<sup>2</sup>). Exposure to high intensity UV light for longer periods may cause permanent damage to the device.

## PROGRAMMING CHARACTERISTICS

Initially, and after erasure, all the EPROM control bits of the 5C032 are connected (in the "1" state). Each of the connected control bits are selectively disconnected by programming the EPROM cells into their "0" state. Programming voltage and waveform specifications are available by request from Intel to support programming of the device.

## Intelligent Programming Algorithm

The 5C032 supports the Intelligent Programming Algorithm which rapidly programs Intel H-ELPDs (and EPROMs) using an efficient and reliable method. The Intelligent Programming Algorithm is particularly suited to the production programming environment.

This method greatly decreases the overall programming time while programming reliability is ensured as the incremental program margin of each bit is continually monitored to determine when the bit has been successfully programmed.

## FUNCTIONAL TESTING

Since the logical operation of the 5C032 is controlled by EPROM elements, the device is completely testable. Each programmable EPROM bit controlling the internal logic is tested using application-independent test program patterns. After testing, the devices are erased before shipment to customers. No post-programming tests of the EPROM array are required.

The testability and reliability of EPROM-based programmable logic devices is an important feature over similar devices based on fuse technology. Fuse-based programmable logic devices require a user to perform post-programming tests to insure proper programming. These tests must be done at the device level because of the cumulative error effect. For example, a board containing ten devices each possessing a 2% device fallout translates into an 18% fallout at the board level (it should be noted that programming fallout of fuse-based programmable logic devices is typically 2% or higher).

## DESIGN RECOMMENDATIONS

For proper operation, it is recommended that all input and output pins be constrained to the voltage range  $GND < (V_{IN} \text{ or } V_{OUT}) < V_{CC}$ . Unused inputs should be tied to an appropriate logic level (e.g. either  $V_{CC}$  or  $GND$ ) to minimize device power consumption. Reserved pins (as indicated in the iPLDS REPORT file) should be left floating (no connect) so that the pin can attain the appropriate logic level. A power supply decoupling capacitor of at least 0.2  $\mu$ F must be connected directly between  $V_{CC}$  and  $GND$  pins of the device.

As with all CMOS devices, ESD handling procedures should be used with the 5C032 to prevent damage to the device during programming, assembly, and test.

## DESIGN SECURITY

A single EPROM bit provides a programmable design security feature that controls the access to the data programmed into the device. If this bit is set, a proprietary design within the device cannot be copied. This EPROM security bit enables a higher degree of design security than fused-based devices since programmed data within EPROM cells is invisi-

ble even to microscopic evaluation. The EPROM security bit, along with all the other EPROM control bits, will be reset by erasing the device.

## AUTOMATIC STAND-BY MODE

The 5C032 contains a programmable bit, the Turbo Bit, that optimizes operation for speed or for power savings. When the Turbo Bit is programmed (TURBO = ON), the device is optimized for maximum speed. When the Turbo bit is not programmed (TURBO = OFF), the device is optimized for power savings by entering standby mode during periods of inactivity.

Figure 4 shows the device entering standby mode approximately 100 ns after the last input transition. When the next input transition is detected, the device returns to active mode. Wakeup time adds an additional 15 ns to the propagation delay through the device as measured from the first input. No delay will occur if an output is dependent on more than one input and the last of the inputs changes after the device has returned to active mode.

After erasure, the Turbo Bit is unprogrammed (OFF); automatic standby mode is enabled. When the Turbo Bit is programmed (ON), the device never enters standby mode.

## LATCH-UP IMMUNITY

All of the input, I/O, and clock pins of the 5C032 have been designed to resist latch-up which is inherent in inferior CMOS structures. The 5C032 is designed with Intel's proprietary CHMOS II-E EPROM

process. Thus, each of the 5C032 pins will not experience latch-up with currents up to  $\pm 100$  mA and voltages ranging from  $-1V$  to  $(V_{CC} + 1V)$ . Furthermore, the programming pin is designed to resist latch-up to the 13.5V maximum device limit.

## SOFTWARE SUPPORT

Full logic compilation and functional simulation for the 5C032 is supported by PLDshell Plus™ software. The GUPI20D20J provides programming support on Intel programmers.

PLDshell Plus design software is Intel's new, user-friendly design tool for  $\mu$ PLD design. PLDshell Plus allows users to incorporate their preferred text editor, programming software, and additional design tools into an easy-to-use, menued design environment that includes Intel's PLDasm™ logic compiler and simulation software along with disassembly, conversion, and translation utilities. The PLDasm compiler and simulator software accepts industry-standard PDS source files that express designs as Boolean equations, truth tables, or state machines. On-line help, datasheet briefs, technical notes, and error message information, along with waveform viewing/printing capability make the design task as easy as possible. PLDshell Plus software is available from Intel Literature channels or from your local Intel sales representative.

Tools that support schematic capture and timing simulation for the 5C032 are available. Support under iPLS II is still available. Please refer to the "Development Tools" section of the Programmable Logic handbook.

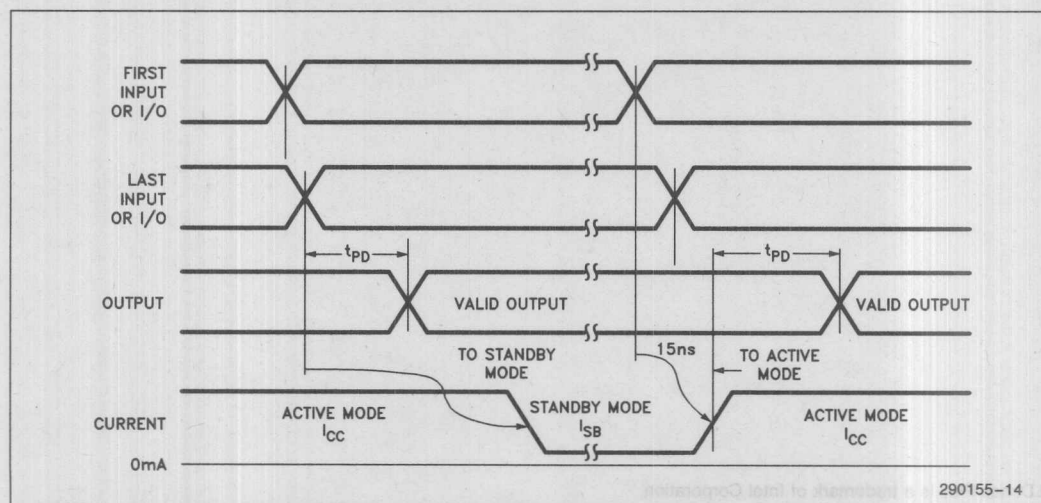


Figure 4. 5C032 Standby and Active Mode Transitions

t <sub>PD</sub> (ns)	t <sub>CO</sub> (ns)	f <sub>MAX</sub> (MHz)	Order Code	Package	Operating Range
30	17	43.5	D5C032-30	CERDIP	Commercial
			P5C032-30	PDIP	
35	20	40	D5C032-35	CERDIP	Commercial
			P5C032-35	PDIP	
40	24	33.3	D5C032-40	CERDIP	Commercial
			P5C032-40	PDIP	

\*ABEL is a trademark of Data I/O, Corp. CUPL is a trademark of Logical Devices, Inc. PLDesigner is a trademark of MINC, Inc. Log/IC is a trademark of ISDATA, Incorporated.

**ABSOLUTE MAXIMUM RATINGS\***

Symbol	Parameter	Min	Max	Unit
$V_{CC}$	Supply Voltage(1)	-2.0	7.0	V
$V_{PP}$	Programming Supply Voltage(1)	-2.0	13.5	V
$V_I$	DC Input Voltage(1)(2)	-0.5	$V_{CC} + 0.5$	V
$t_{stg}$	Storage Temperature	-65	+150	°C
$t_{amb}$	Ambient Temperature(4)	-10	+85	°C

**NOTES:**

1. Voltages with respect to ground.
2. Minimum DC input is -0.5V. During transitions, the inputs may undershoot to -2.0V or overshoot to +7.0V for periods less than 20 ns under no load conditions.
3. Under bias, Extended temperature versions are also available.
4. Extended temperature versions also available.

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**\*WARNING:** Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.

2

**RECOMMENDED OPERATING CONDITIONS**

Symbol	Parameter	Min	Max	Unit
$V_{CC}$	Supply Voltage	4.75	5.25	V
$V_{IN}$	Input Voltage	0	$V_{CC}$	V
$V_O$	Output Voltage	0	$V_{CC}$	V
$T_A$	Operating Temperature	0	+70	°C
$t_R$	Input Rise Time		500	ns
$t_F$	Input Fall Time		500	ns

**PACKAGE/TECHNOLOGY SPECIFICATIONS**

Description	Specification
$\theta_{JA}$ —Junction-to-Ambient Thermal Resistance	83°C/W—CerDIP 109°C/W—PDIP
$\theta_{JC}$ —Junction-to-Case Thermal Resistance	20°C/W—CerDIP 20°C/W—PDIP
$I_{CC}$ Hot—Ambient @ 70°C	30 mA
$I_{CC}$ Typical—Ambient @ 25°C	30 mA
Process	CHMOS IIE, PX24

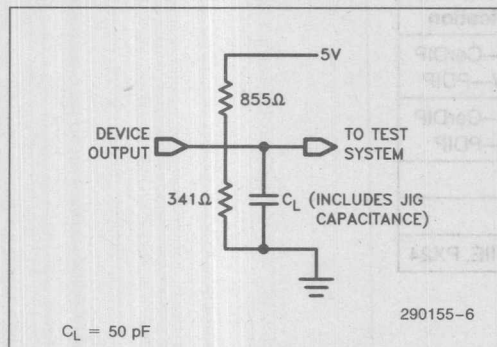
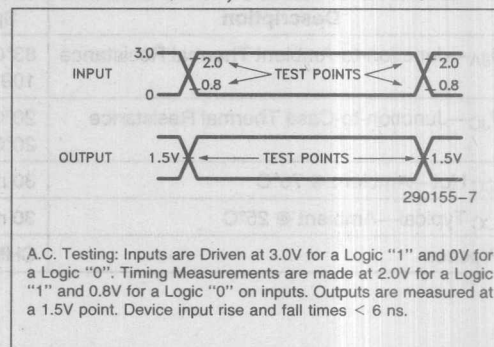


**D.C. CHARACTERISTICS**  $T_A = 0^\circ\text{C to } 70^\circ\text{C}$ ,  $V_{CC} = 5V \pm 5\%$ 

Symbol	Parameter/Test Conditions	Min	Typ	Max	Unit
$V_{IH}^{(5)}$	High Level Input Voltage	2.0		$V_{CC} + 0.3$	V
$V_{IL}^{(5)}$	Low Level Input Voltage	-0.3		0.8	V
$V_{OH}^{(6)}$	High Level Output Voltage $I_O = -4.0\text{ mA D.C.}, V_{CC} = \text{min.}$	2.4			V
$V_{OL}$	Low Level Output Voltage $I_O = 4.0\text{ mA D.C.}, V_{CC} = \text{min.}$			0.45	V
$I_I$	Input Leakage Current $V_{CC} = \text{max.}, GND < V_{IN} < V_{CC}$			$\pm 10$	$\mu\text{A}$
$I_{OZ}$	Output Leakage Current $V_{CC} = \text{max.}, GND < V_{OUT} < V_{CC}$			$\pm 10$	$\mu\text{A}$
$I_{SC}^{(7)}$	Output Short Circuit Current $V_{CC} = \text{max.}, V_{OUT} = 0.5V$			10	mA
$I_{SB}^{(8)}$	Standby Current $V_{CC} = \text{max.}, V_{IN} = V_{CC} \text{ or } GND$ , Standby Mode		10	100	$\mu\text{A}$
$I_{CC}^{(9)}$	Power Supply Current $V_{CC} = \text{max.}, V_{IN} = V_{CC} \text{ or } GND$ , No Load, Input Freq. = 10 MHz Active Mode (Turbo = Off), Device Prog. as 8-bit Ctr.		15	25	mA

**NOTES:**

5. Absolute values with respect to device GND; all over- and undershoots due to system or tester noise are included.
6.  $I_O$  at CMOS levels (3.84V) = -2 mA.
7. Not more than 1 output should be tested at a time. Duration of that test must not exceed 1 second.
8. With Turbo Bit = Off, device automatically enters standby mode approximately 100 ns after last input transition.
9. Maximum Active Current at operational frequency is less than 40 mA.

**A.C. TESTING LOAD CIRCUIT****A.C. TESTING INPUT, OUTPUT WAVEFORM**

## CAPACITANCE

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
C <sub>IN</sub>	Input Capacitance	V <sub>IN</sub> = 0V, f = 1.0 MHz			10	pF
C <sub>OUT</sub>	Output Capacitance	V <sub>OUT</sub> = 0V, f = 1.0 MHz			10	pF
C <sub>CLK</sub>	Clock Pin Capacitance	V <sub>IN</sub> = 0V, f = 1.0 MHz			10	pF
C <sub>VPP</sub>	V <sub>pp</sub> Pin	Pin 11, f = 1.0 MHz			20	pF

## A.C. CHARACTERISTICS T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5V ± 5%, Turbo Bit On<sup>(10)</sup>

Symbol	From	To	5C032-30			5C032-35			5C032-40			Non-(8) Turbo Mode	Unit
			Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
t <sub>PD</sub>	I or I/O	Comb. Output			30			35			40	+ 15	ns
t <sub>PZX</sub> <sup>(11)</sup>	I or I/O	Output Enable			30			35			40	+ 15	ns
t <sub>PXZ</sub> <sup>(11)</sup>	I or I/O	Output Disable			30			35			40	+ 15	ns

### NOTES:

10. Typ. values are at T<sub>A</sub> = 25°C, V<sub>CC</sub> = 5V, Active Mode.

11. t<sub>PZX</sub> and t<sub>PXZ</sub> are measured at ±0.5V from steady state voltage as driven by spec. output load. t<sub>PXZ</sub> is measured with C<sub>L</sub> = 5 pF.

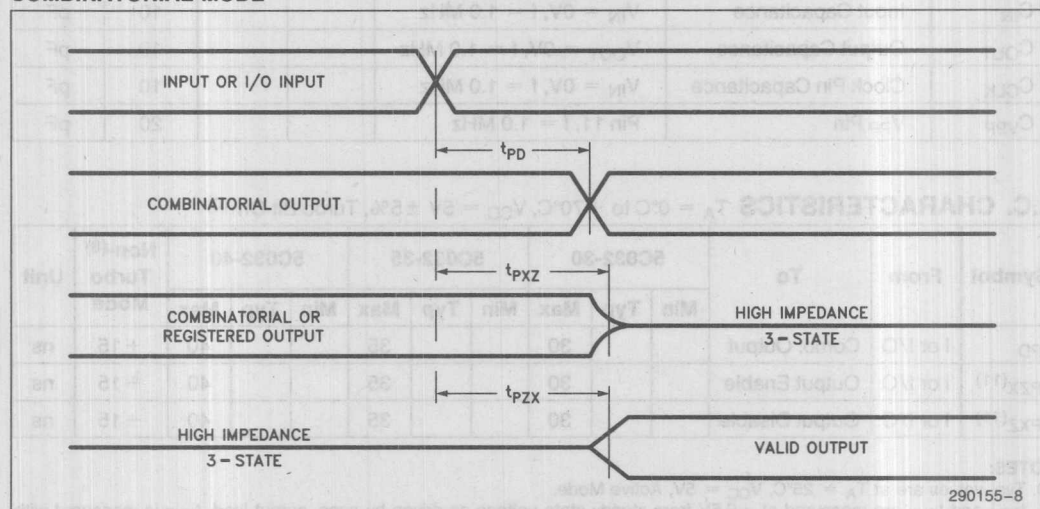
## A.C. CHARACTERISTICS T<sub>A</sub> = 0°C to 70°C, V<sub>CC</sub> = 5V ± 5%, Turbo Bit On<sup>(10)</sup>

### SYNCHRONOUS CLOCK MODE

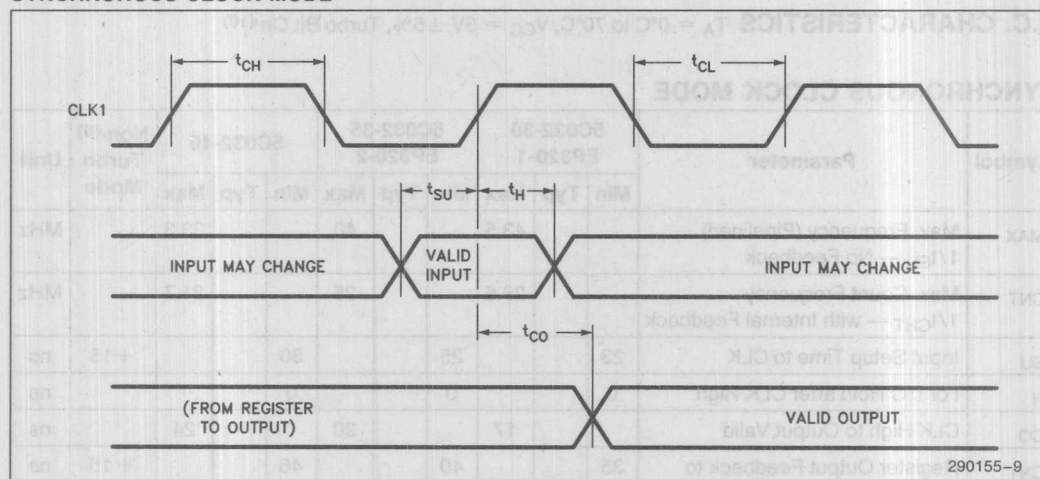
Symbol	Parameter	5C032-30 EP320-1			5C032-35 EP320-2			5C032-40			Non-(8) Turbo Mode	Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
f <sub>MAX</sub>	Max. Frequency (Pipelined) 1/t <sub>SU</sub> — No Feedback			43.5			40			33.3		MHz
f <sub>CNT</sub>	Max. Count Frequency 1/t <sub>CNT</sub> — with Internal Feedback			28.5			25			21.7		MHz
t <sub>SU</sub>	Input Setup Time to CLK	23			25			30			+ 15	ns
t <sub>H</sub>	I or I/O Hold after CLK High	0			0			0				ns
t <sub>CO</sub>	CLK High to Output Valid			17			20			24		ns
t <sub>CNT</sub>	Register Output Feedback to Register Input — Internal Path	35			40			46			+ 15	ns
t <sub>CH</sub>	CLK High Time	11			12			15				ns
t <sub>CL</sub>	CLK Low Time	11			12			15				ns

## SWITCHING WAVEFORMS

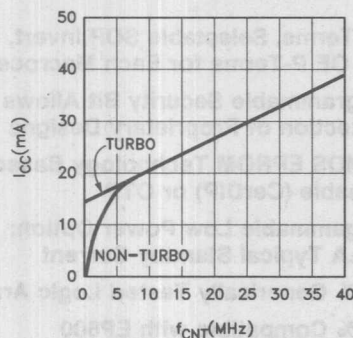
## COMBINATORIAL MODE



## SYNCHRONOUS CLOCK MODE



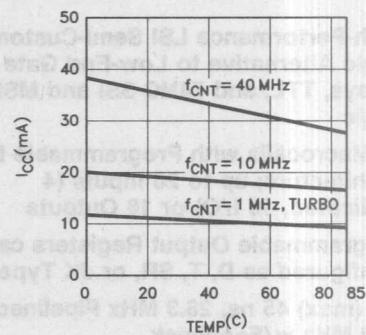
Current in Relation to Frequency



Conditions:  $T_A = 0^\circ\text{C}$ ,  $V_{CC} = 5.25\text{V}$

290155-11

Current in Relation to Temperature

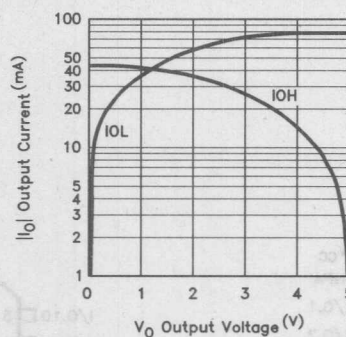


Conditions:  $V_{CC} = 5.25\text{V}$

290155-12

2

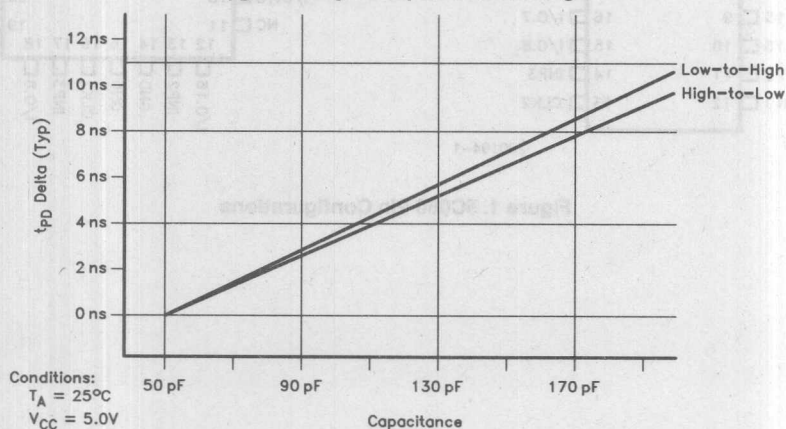
Output Drive Current in Relation to Voltage



Conditions:  $T_A = +25^\circ\text{C}$ ,  $V_{CC} = 5\text{V}$

290155-13

$t_{PD}$  Derating vs Capacitive Loading



Conditions:  
 $T_A = 25^\circ\text{C}$   
 $V_{CC} = 5.0\text{V}$

Capacitance

290155-15

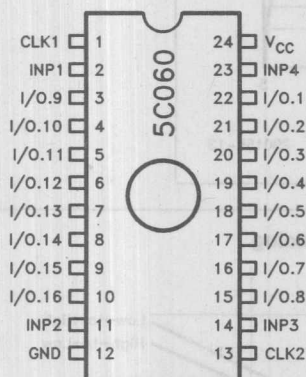


# 5C060

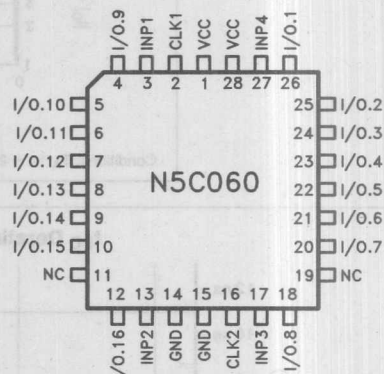
## 16-MACROCELL CHMOS EPLD

- High-Performance LSI Semi-Custom Logic Alternative to Low-End Gate Arrays, TTL, and 74HC SSI and MSI Logic
- 16 Macrocells with Programmable I/O Architecture; up to 20 Inputs (4 Dedicated, 16 I/O) or 16 Outputs
- Programmable Output Registers can be Configured as D, T, SR, or JK Types
- $t_{PD}$  (max) 45 ns, 26.3 MHz Pipelined, 22.2 MHz w/Feedback
- Programmable Clock System with 2 Synchronous Clocks and Asynchronous Clocking Option on all Registers
- 8 P-Terms, Selectable SOP Invert, Clear and OE P-Terms for Each Macrocell
- Programmable Security Bit Allows Total Protection of Proprietary Designs
- CHMOS EPROM Technology Based. UV Erasable (CerDIP) or OTP
- Programmable Low Power Option; 50  $\mu$ A Typical Standby Current
- 100% Generically Tested Logic Array
- 100% Compatible with EP600
- Available in 24-Pin 300-mil CerDIP/PDIP and 28-Pin PLCC Packages

(See Packaging Specifications, Order Number 240800, Package Types D, P, and N)



290194-1



290194-2

Figure 1. 5C060 Pin Configurations

The Intel 5C060 is a 16-macrocell, 24-pin, general purpose device. The device can be used to replace low-end gate arrays, multiple programmable logic arrays and LS TTL and 74HC (CMOS) SSI and MSI logic devices. The 5C060 can also be used as a direct,

low power replacement for many 24-pin fuse-based programmable logic devices. With its revolutionary programmable I/O architecture, the device has advanced functional capabilities beyond that of typical programmable logic. Figure 2 shows the global architecture of the device.

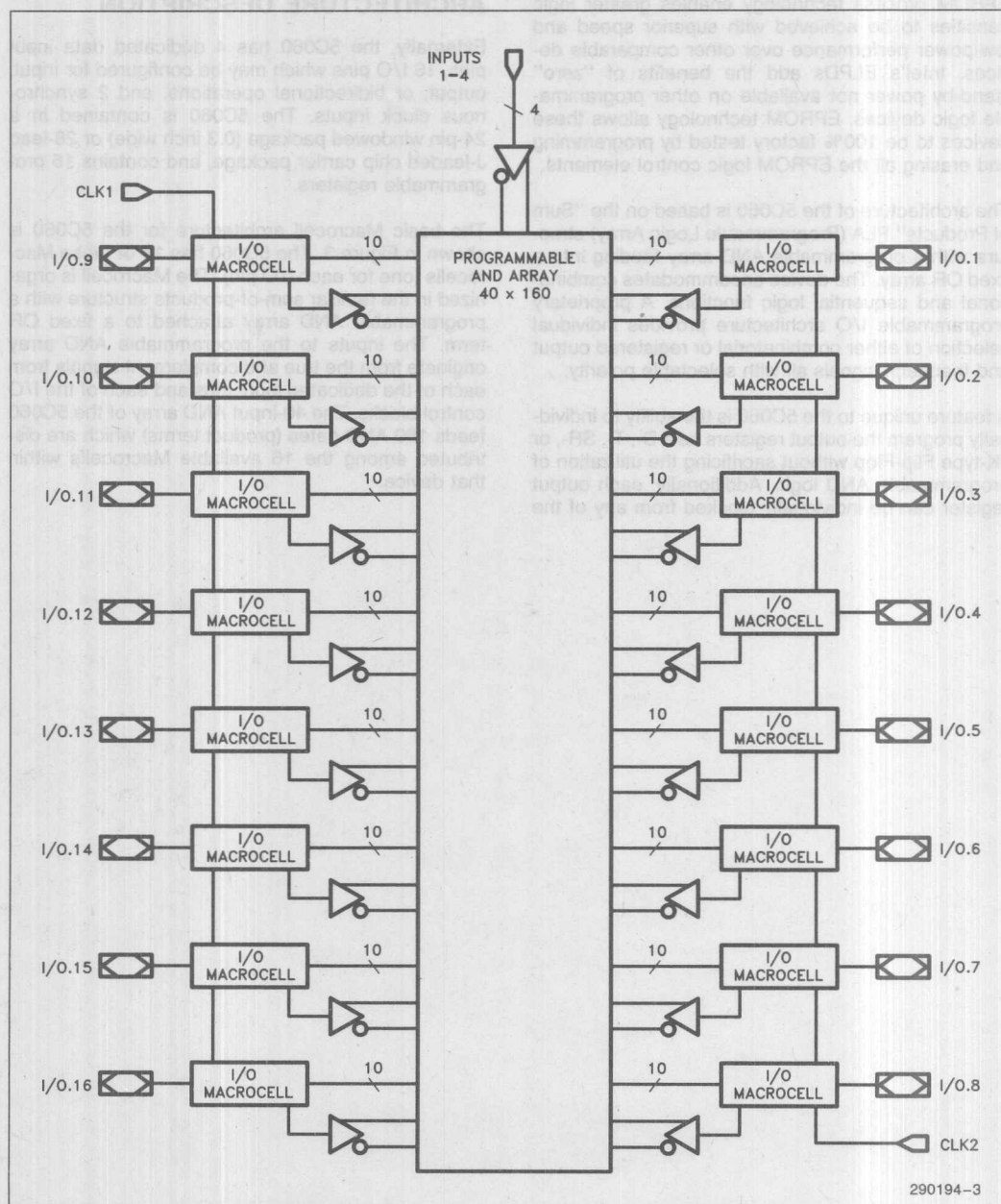


Figure 2. 5C060 Global Architecture

The 5C060 EPLD uses CHMOS EPROM (floating gate) cells as logic control elements instead of fuses. The CHMOS EPROM technology reduces power consumption of EPLDs to less than 20% of a comparable bipolar device without sacrificing speed performance. In addition, Intel's advanced CHMOS II-E EPROM process technology enables greater logic densities to be achieved with superior speed and low-power performance over other comparable devices. Intel's EPLDs add the benefits of "zero" stand-by power not available on other programmable logic devices. EPROM technology allows these devices to be 100% factory tested by programming and erasing all the EPROM logic control elements.

The architecture of the 5C060 is based on the "Sum of Products" PLA (Programmable Logic Array) structure with a programmable AND array feeding into a fixed OR array. The device accommodates combinational and sequential logic functions. A proprietary programmable I/O architecture provides individual selection of either combinatorial or registered output and feedback signals all with selectable polarity.

A feature unique to the 5C060 is the ability to individually program the output registers as a D-, T-, SR-, or JK-type Flip-Flop without sacrificing the utilization of programmable AND logic. Additionally, each output register can be individually clocked from any of the

input or feedback paths available within the AND array. With these features, a wide variety of logic functions can be simultaneously implemented—all on the same device.

## ARCHITECTURE DESCRIPTION

Externally, the 5C060 has 4 dedicated data input pins, 16 I/O pins which may be configured for input, output, or bidirectional operations, and 2 synchronous clock inputs. The 5C060 is contained in a 24-pin windowed package (0.3 inch wide) or 28-lead J-leaded chip carrier package, and contains 16 programmable registers.

The basic Macrocell architecture for the 5C060 is shown in Figure 3. The 5C060 has 16 of these Macrocells (one for each I/O pin). The Macrocell is organized in the familiar sum-of-products structure with a programmable AND array attached to a fixed OR term. The inputs to the programmable AND array originate from the true and complement signals from each of the dedicated input pins and each of the I/O control blocks. The 40-input AND array of the 5C060 feeds 160 AND gates (product terms) which are distributed among the 16 available Macrocells within that device.

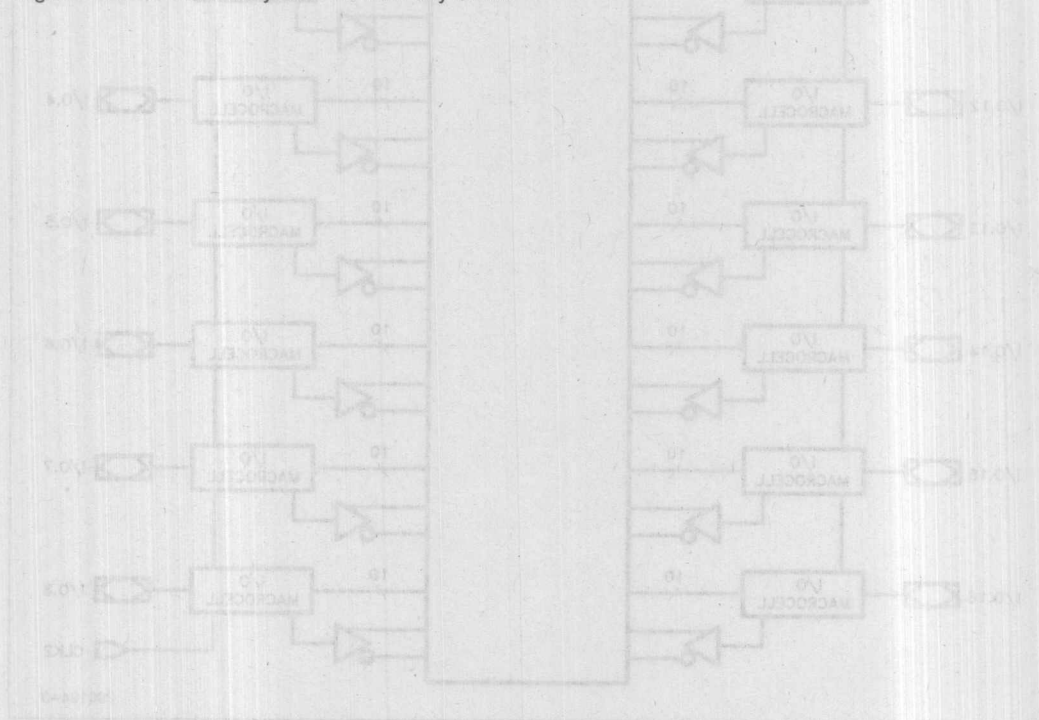


Figure 3. 5C060 Macrocell Architecture

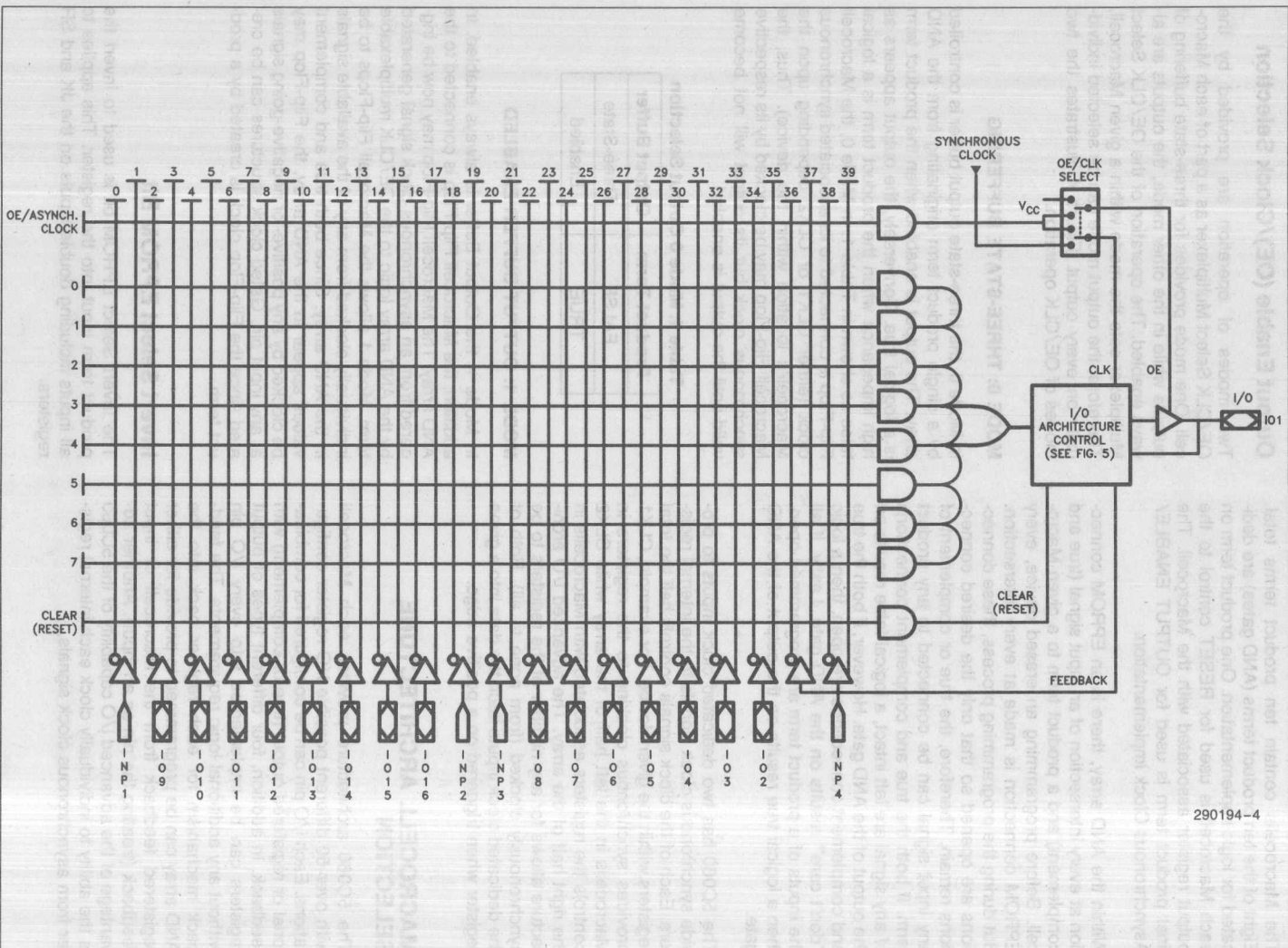


Figure 3. 5C060 Macrocell Architecture



The Macrocells contain ten product terms total. Eight of the ten product terms (AND gates) are dedicated for logic implementation. One product term on each Macrocell is used for RESET control to the output register associated with the Macrocell. The final product term is used for OUTPUT ENABLE/Asynchronous Clock implementation.

Within the AND array, there is an EPROM connection at every intersection of an input signal (true and complement) and a product term to a given Macrocell. Before programming an erased device, every EPROM connection is made at every intersection. But during the programming process, these connections are opened so that only the desired connections remain. Therefore, the true or complement of any input signal can be connected to any product term. If both the true and complement connections of any signal are left intact, a logical false results on the output of the AND gate. However, if both the true and complement connections are open, then a logic "don't care" results on the AND gate. Lastly, if all the inputs of a product term are programmed open, then a logical true results on the output of the AND gate.

The 5C060 has two dedicated clock inputs to provide synchronous clock signals to the internal registers. Each of the clock signals controls half the total registers within the given device. For example, CLK1 provides synchronous clocking to the registers in Macrocells in the left half of the array while CLK2 controls the registers associated with Macrocells in the right half of the array. The advanced I/O architecture allows for any number of the registers to be synchronously clocked (from none to all). Both of the dedicated clock inputs latch the data into a given register when triggered on a positive edge.

## MACROCELL ARCHITECTURE SELECTION

The 5C060 architecture provides each Macrocell with over 50 different possible I/O register configurations. Each I/O pin can be configured for combinatorial or registered output (true or complement) with feedback. In addition, four different types of output registers can be implemented into every I/O pin without any additional logic requirements. The feedback mechanism for each register back into the AND array can be programmed to provide for either registered feedback from the Macrocell or input feedback (treating the pin as an input). Another advantage of the advanced I/O capability of the 5C060 is the ability to individually clock each internal register from asynchronous clock signals.

## Output Enable (OE)/Clock Selection

Two modes of operation are provided by the OE/CLK Select Multiplexer as a part of each Macrocell. One mode provides for three-state buffering of outputs while in the other mode, the outputs are always enabled. The operation of the OE/CLK Select Multiplexer sets the mode within a given Macrocell. Therefore, the output mode can be selected individually on every output. Figure 4 illustrates the two modes of OE/CLK operation.

### MODE 0: THREE-STATE BUFFERING

In Mode 0, the three-state output buffer is controlled by a single product term originating from the AND array. The output is enabled when the product term is a logical true. Conversely, the output appears as high impedance when the product term is a logical false as shown in Table 1. In Mode 0, the Macrocell Flip-Flop is connected to its associated synchronous clock (either CLK1 or CLK2 depending upon the Macrocell's location within the device). Thus, the Macrocell Flip-Flop may be clocked by its respective synchronous clock but its output will not become valid until the output is enabled.

Table 1. Mode 0 Output Selection

Product Term	Output Buffer
FALSE	Three-State
TRUE	Enabled

### MODE 1: OUTPUT BUFFER ENABLED

In Mode 1, the Output Buffer is always enabled. In addition, the Macrocell Flip-Flop is connected to the AND array. The Macrocell Flip-Flop may now be triggered from an asynchronous clock signal generated by the AND array logic to the OE/CLK multiplexable term. Mode 1 allows the Macrocell Flip-Flops to be individually clocked from any of the available signals in the AND array. Since both true and complement values appear in the AND array, the Flip-Flop may be clocked by any positive- or negative-going signals at any input pin. Gated clock structures can be created since the Flip-Flop clock is created by a product term.

## Invert Select EPROM Bit

The Invert Select EPROM bit is used to invert the product term input into the register. This applies to all inputs including double inputs on the JK and SR registers.

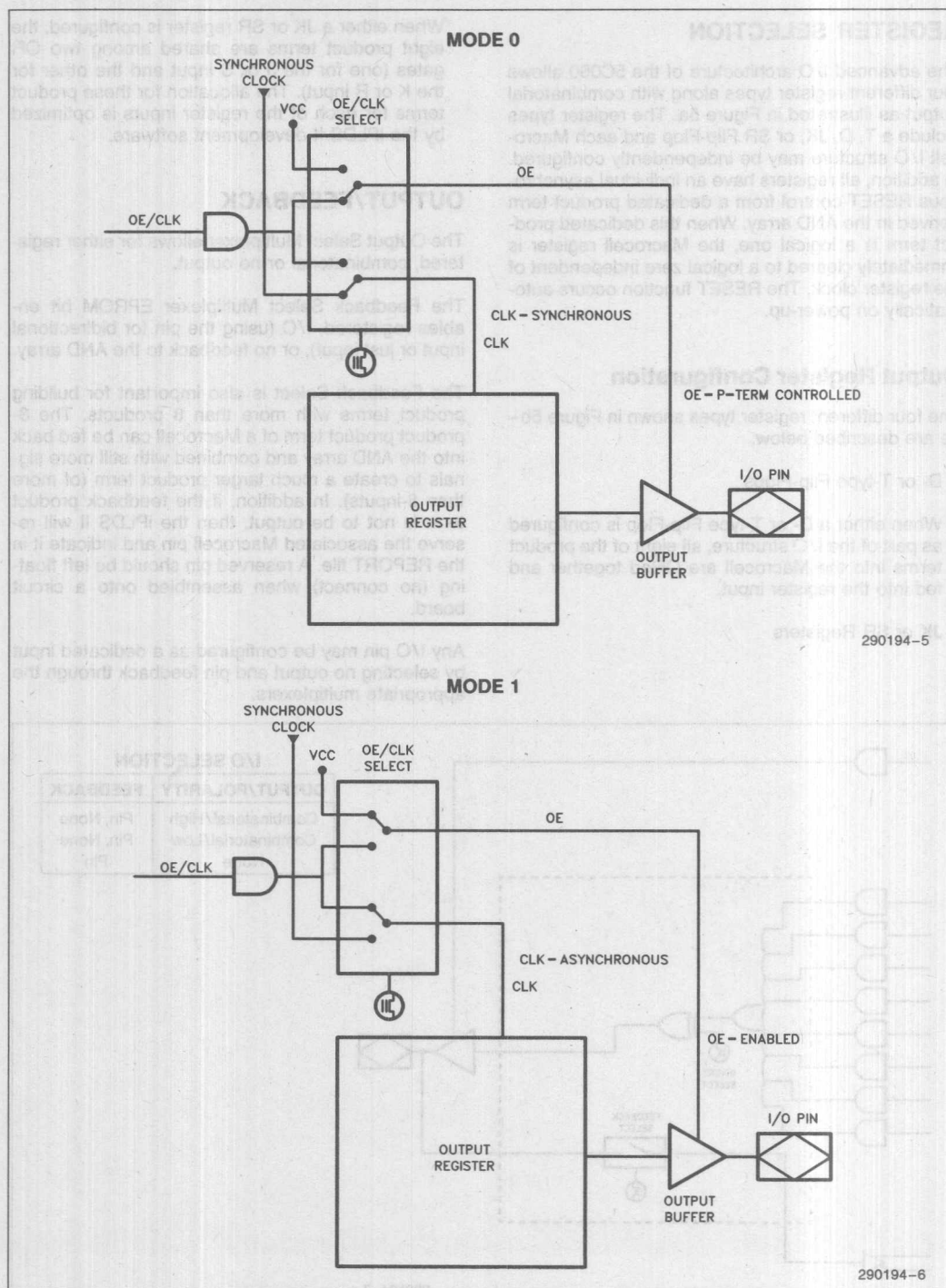


Figure 4. Output Enable/Clock Configuration

## REGISTER SELECTION

The advanced I/O architecture of the 5C060 allows four different register types along with combinatorial output as illustrated in Figure 5a. The register types include a T, D, JK, or SR Flip-Flop and each Macrocell I/O structure may be independently configured. In addition, all registers have an individual asynchronous RESET control from a dedicated product term derived in the AND array. When this dedicated product term is a logical one, the Macrocell register is immediately cleared to a logical zero independent of the register clock. The RESET function occurs automatically on power-up.

## Output Register Configuration

The four different register types shown in Figure 5b-5e are described below.

### D- or T-type Flip-Flops

When either a D- or T-type Flip-Flop is configured as part of the I/O structure, all eight of the product terms into the Macrocell are ORed together and fed into the register input.

### JK or SR Registers

When either a JK or SR register is configured, the eight product terms are shared among two OR gates (one for the J or S input and the other for the K or R input). The allocation for these product terms for each of the register inputs is optimized by the iPLDS II development software.

## OUTPUT/FEEDBACK

The Output Select Multiplexer allows for either registered, combinatorial or no output.

The Feedback Select Multiplexer EPROM bit enables registered, I/O (using the pin for bidirectional input or just input), or no feedback to the AND array.

The Feedback Select is also important for building product terms with more than 8 products. The 8-product product term of a Macrocell can be fed back into the AND array and combined with still more signals to create a much larger product term (of more than 8-inputs). In addition, if the feedback product term is not to be output, then the iPLDS II will reserve the associated Macrocell pin and indicate it in the REPORT file. A reserved pin should be left floating (no connect) when assembled onto a circuit board.

Any I/O pin may be configured as a dedicated input by selecting no output and pin feedback through the appropriate multiplexers.

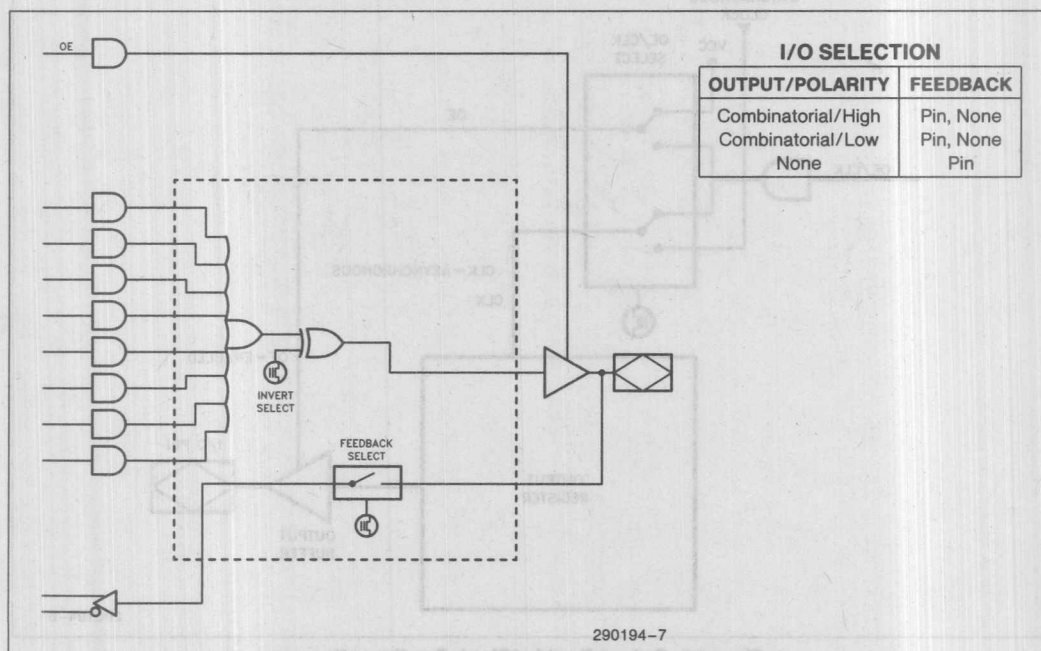
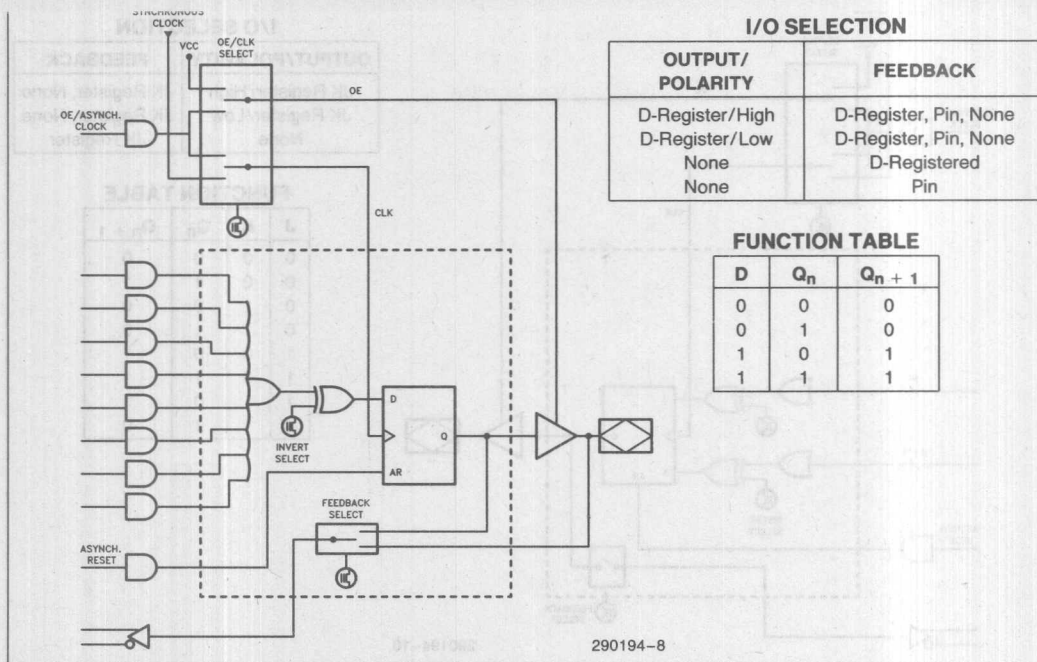
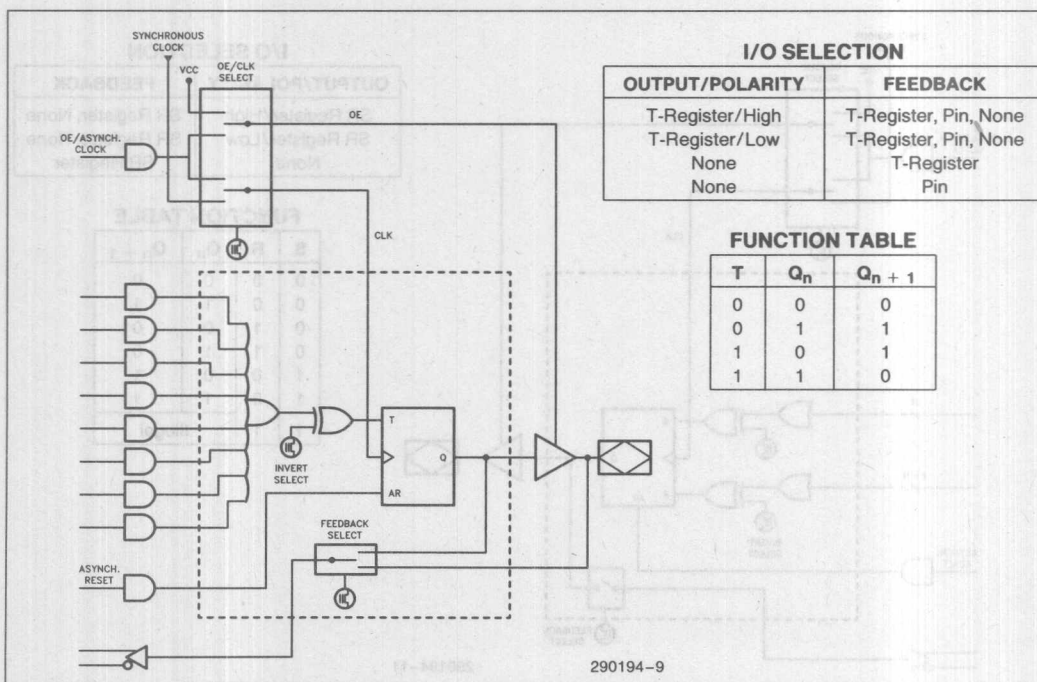


Figure 5a. Combinatorial I/O Configuration



**Figure 5b. D-Type Flip-Flop Register Configuration**



**Figure 5c. Toggle Flip-Flop Register Configuration**



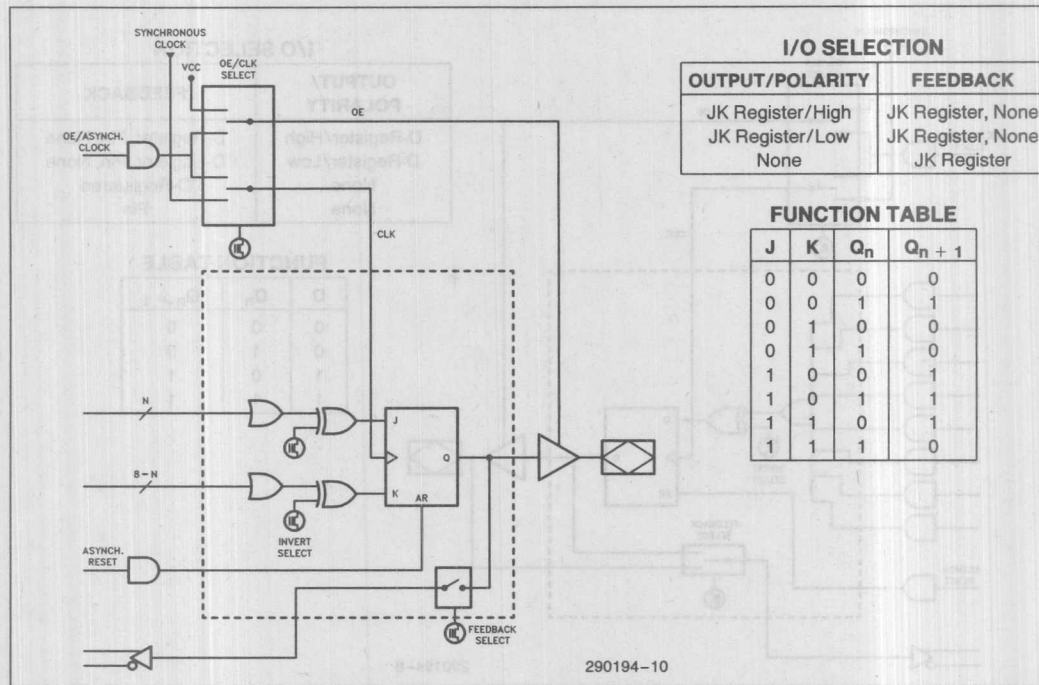


Figure 5d. JK Flip-Flop Register Configuration

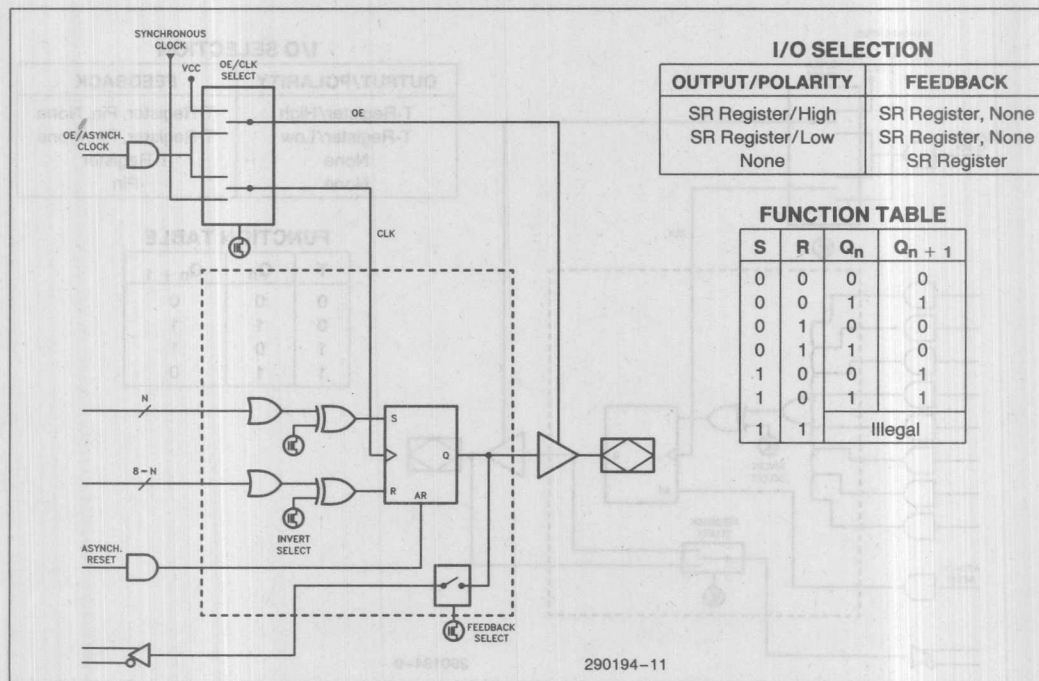


Figure 5e. SR Flip-Flop Register Configuration

## Erased-State Configuration

Prior to programming or after erasing, the I/O structure is configured for combinatorial active low output with input (pin) feedback.

## ERASURE CHARACTERISTICS

Erasure characteristics of the device are such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000Å. It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000Å–4000Å. Data shows that constant exposure to room level fluorescent lighting could erase the typical device in approximately three years, while it would take approximately one week to cause erasure when exposed to direct sunlight. If the 5C060 is to be exposed to these types of lighting conditions for extended periods of time, conductive opaque labels should be placed over the device window to prevent unintentional erasure.

The recommended erasure procedure for the 5C060 is exposure to shortwave ultraviolet light with a wavelength of 2537Å. The integrated dose (i.e., UV intensity  $\times$  exposure time) for erasure should be a minimum of fifteen (15) Wsec/cm<sup>2</sup>. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12,000  $\mu$ W/cm<sup>2</sup> power rating. The 5C060 should be placed within one inch of the lamp tubes during erasure. The maximum integrated dose the 5C060 can be exposed to without damage is 7258 Wsec/cm<sup>2</sup> (1 week at 12,000  $\mu$ W/cm<sup>2</sup>). Exposure to high intensity UV light for longer periods may cause permanent damage to the device.

## PROGRAMMING CHARACTERISTICS

Initially, and after erasure, all the EPROM control bits of the 5C060 are connected (in the "1" state). Each of the connected control bits are selectively disconnected by programming the EPROM cells into their "0" state. Programming voltage and waveform specifications are available by request from Intel to support programming of the 5C060.

## Intelligent Programming Algorithm

The 5C060 supports the Intelligent Programming Algorithm which rapidly programs Intel ELPDs using an efficient and reliable method. The Intelligent Programming Algorithm is particularly suited to the production programming environment. This method ensures reliability as the incremental program margin of each bit is continually monitored to determine when the bit has been successfully programmed.

## FUNCTIONAL TESTING

Since the logical operation of the 5C060 is controlled by EPROM elements, the device is completely testable. Each programmable EPROM bit controlling the internal logic is tested using application-independent test program patterns. After testing, the devices are erased before shipment to customers. No post-programming tests of the EPROM array are required.

The testability and reliability of EPROM-based programmable logic devices is an important feature over similar devices based on fuse technology. Fuse-based programmable logic devices require a user to perform post-programming tests to insure proper programming. These tests must be done at the device level because of the cumulative error effect. For example, a board containing ten devices each possessing a 2% device fallout translates into an 18% fallout at the board level (it should be noted that programming fallout of fuse-based programmable logic devices is typically 2% or higher).

## DESIGN RECOMMENDATIONS

For proper operation, it is recommended that all input and output pins be constrained to the voltage range  $GND < (V_{IN} \text{ or } V_{OUT}) < V_{CC}$ . Unused inputs and I/Os should be tied to  $V_{CC}$  or GND to minimize device power consumption. Reserved pins (as indicated in the logic compiler REPORT file) should be left floating (no connect) so that the pin can attain the appropriate logic level. A power supply decoupling capacitor of at least 0.2  $\mu$ F must be connected directly between  $V_{CC}$  and GND pins of the device.

As with all CMOS devices, ESD handling procedures should be used with the 5C060 to prevent damage to the device during programming, assembly, and test.

## DESIGN SECURITY

A single EPROM bit provides a programmable design security feature that controls the access to the data programmed into the device. If this bit is set, a proprietary design within the device cannot be copied. This EPROM security bit enables a higher degree of design security than fused-based devices

since programmed data within EPROM cells is invisible even to microscopic evaluation. The EPROM security bit, along with all the other EPROM control bits, will be reset by erasing the device.

### AUTOMATIC STAND-BY MODE

The 5C060 contains a programmable bit, the Turbo Bit, that optimizes operation for speed or for power savings. When the Turbo Bit is programmed (TURBO = ON), the device is optimized for maximum speed. When the Turbo Bit is not programmed (TURBO = OFF), the device is optimized for power savings by entering standby mode during periods of inactivity.

Figure 6 shows the device entering standby mode approximately 100 ns after the last input transition. When the next input transition is detected, the device returns to active mode. Wakeup time adds an additional 25 ns to the propagation delay through the device as measured from the first input. No delay will occur if an output is dependent on more than one input and the last of the inputs changes after the device has returned to active mode.

After erasure, the Turbo Bit is unprogrammed (OFF); automatic standby mode is enabled. When the Turbo Bit is programmed (ON), the device never enters standby mode.

### LATCH-UP IMMUNITY

All of the input, I/O, and clock pins of the 5C060 have been designed to resist latch-up which is inher-

ent in inferior CMOS structures. The 5C060 is designed with Intel's proprietary CHMOS II-E EPROM process. Thus, each of the pins will not experience latch-up with currents up to  $\pm 100$  mA and voltages ranging from  $-1V$  to  $(V_{CC} + 1V)$ . Furthermore, the programming pin is designed to resist latch-up to the 13.5V maximum device limit.

### SOFTWARE SUPPORT

Full logic compilation and functional simulation for the 5C060 is supported by PLDshell Plus™ software. The GUPI Logic-11D provides programming support on Intel programmers.

PLDshell Plus™ design software is Intel's new, user-friendly design tool for  $\mu$ PLD design. PLDshell Plus allows users to incorporate their preferred text editor, programming software, and additional design tools into an easy-to-use, menued design environment that includes Intel's PLDasm™ logic compiler and simulation software along with disassembly, conversion, and translation utilities. The PLDasm compiler and simulator software accepts industry-standard PDS source files that express designs as Boolean equations, truth tables, or state machines. On-line help, data sheet briefs, technical notes, and error message information, along with waveform viewing/printing capability make the design task as easy as possible. PLDshell Plus software is available from Intel Literature channels or from your local Intel sales representative.

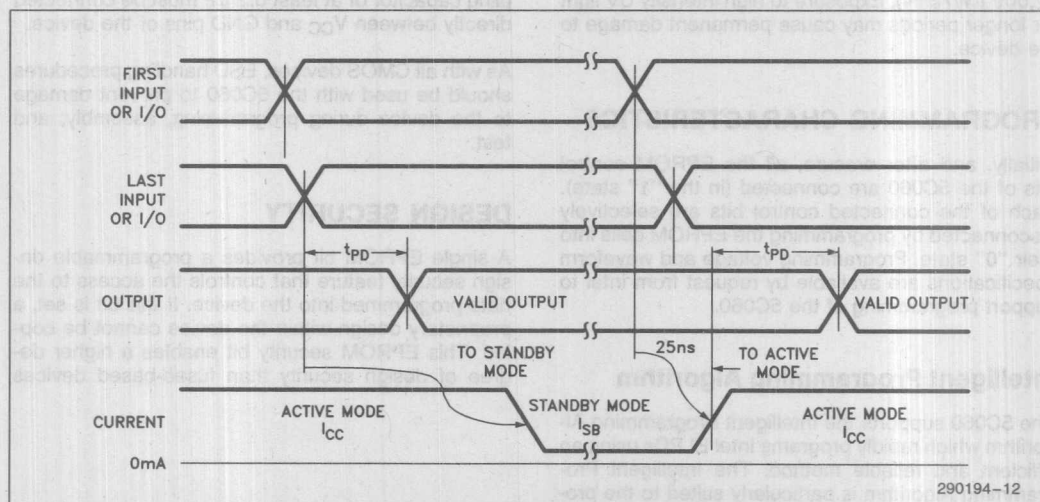


Figure 6. 5C060 Standby and Active Mode Transitions

\*PLDshell Plus™ is a trademark of Intel Corporation.

simulation for the 5C060 are available. Support under iPLS II is still available. Please refer to the "Development Tools" section of the Programmable Logic Handbook.

The 5C060 is also supported by third-party logic compilers such as ABEL\*, CUPL\*, PLDesigner\*, Log/IC\*, etc. Programming support is provided by third-party programmer companies such as Data I/O, Logical Devices, STAG, etc. Please refer to the "Third-Party Support" lists in the *Programmable Logic* handbook for complete information and vendor contacts.

## ORDERING INFORMATION

t <sub>PD</sub> (ns)	t <sub>CO</sub> (ns)	f <sub>MAX</sub> (MHz)	Order Code	Package	Operating Range
45	22	26	D5C060-45	CERDIP	Commercial
			P5C060-45	PDIP	
			N5C060-45	PLCC	
55	25	23	D5C060-55	CERDIP	Commercial
			P5C060-55	PDIP	
			N5C060-55	PLCC	

2

Symbol	Parameter	Min	Max	Unit
V <sub>CC</sub>	Supply Voltage	4.75	5.25	V
V <sub>IN</sub>	Input Voltage	0	V <sub>CC</sub>	V
V <sub>O</sub>	Output Voltage	0	V <sub>CC</sub>	V
T <sub>A</sub>	Operating Temperature	0	+70	°C
t <sub>PLH</sub>	Input Rise Time		800	ns
t <sub>PLL</sub>	Input Fall Time		800	ns

NOTE:  
A. t<sub>PLH</sub> < t<sub>PLL</sub> < 250 ns max.

## PACKAGED TECHNOLOGY SPECIFICATIONS

Description	Specification
θ <sub>JA</sub> —Junction to Ambient Thermal Resistance	84°C/W—CERDIP 87°C/W—PDIP 87°C/W—PLCC
θ <sub>JC</sub> —Junction to Case Thermal Resistance	17°C/W—CERDIP 22°C/W—PDIP 20°C/W—PLCC
T <sub>CH</sub> —Hot—Ambient @70°C	80 mA
T <sub>CD</sub> —Cold—Ambient @25°C	80 mA
Process	CMOS II, 1.5 μm

## D.C. CHARACTERISTICS T<sub>A</sub> = 0°C to 70°C, V<sub>CC</sub> = 5.0V ± 5%

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V <sub>IH</sub> (1)	HIGH Level Input Voltage		2.0		V <sub>CC</sub> + 0.3	V
V <sub>IL</sub> (2)	LOW Level Input Voltage		-0.3		0.8	V
V <sub>OH</sub> (3)	HIGH Level Output Voltage	I <sub>O</sub> = -4.0 mA DC, V <sub>CC</sub> = Min	2.4			V
V <sub>OL</sub> (4)	LOW Level Output Voltage	I <sub>O</sub> = 4.0 mA DC, V <sub>CC</sub> = Min			0.45	V
I <sub>IL</sub> (5)	Input Leakage Current	V <sub>CC</sub> = Max, GND < V <sub>IL</sub> < V <sub>CC</sub>			±10.0	μA
I <sub>OL</sub> (6)	Output Leakage Current	V <sub>CC</sub> = Max, GND < V <sub>OL</sub> < V <sub>CC</sub>			±10.0	μA
I <sub>SC</sub> (7)	Output Short Circuit Current	V <sub>CC</sub> = Max, V <sub>OUT</sub> = 0.8V	50		70	mA

\*Abel is a trademark of Data I/O, Corp. CUPL is a trademark of Logical Devices, Inc. PLDesigner is a trademark of MINC, Inc. Log/IC is a trademark of ISDATA, Inc.



**ABSOLUTE MAXIMUM RATINGS\***

Symbol	Parameter	Min	Max	Units
V <sub>CC</sub>	Supply Voltage <sup>(1)</sup>	-2.0	7.0	V
V <sub>PP</sub>	Programming Supply Voltage <sup>(1)</sup>	-2.0	13.5	V
V <sub>I</sub>	DC Input Voltage <sup>(1)(2)</sup>	-0.5	V <sub>CC</sub> + 0.5	V
t <sub>stg</sub>	Storage Temperature	-65	+150	°C
t <sub>amb</sub>	Ambient Temperature <sup>(3)</sup>	-10	+85	°C

**NOTES:**

1. Voltages with respect to ground.
2. Minimum DC input is -0.5V. During transitions, the inputs may undershoot to -2.0V or overshoot to 7.0V for periods less than 20 ns under no load conditions.
3. Under bias. Extended temperature versions are also available.

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

*\*WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

**RECOMMENDED OPERATING CONDITIONS**

Symbol	Parameter	Min	Max	Unit
V <sub>CC</sub>	Supply Voltage	4.75	5.25	V
V <sub>IN</sub>	Input Voltage	0	V <sub>CC</sub>	V
V <sub>O</sub>	Output Voltage	0	V <sub>CC</sub>	V
T <sub>A</sub>	Operating Temperature	0	+70	°C
t <sub>R</sub> <sup>(4)</sup>	Input Rise Time		500	ns
t <sub>F</sub> <sup>(4)</sup>	Input Fall Time		500	ns

**NOTE:**

4. t<sub>R</sub>, t<sub>F</sub> for CLK is 250 ns max.

**PACKAGE/TECHNOLOGY SPECIFICATIONS**

Description	Specification
θ <sub>JA</sub> —Junction-to-Ambient Thermal Resistance	54°C/W—CerDIP 67°C/W—PDIP 61°C/W—PLCC
θ <sub>JC</sub> —Junction-to-Case Thermal Resistance	17°C/W—CerDIP 22°C/W—PDIP 20°C/W—PLCC
I <sub>CC</sub> Hot—Ambient @70°C	80 mA
I <sub>CC</sub> Typical—Ambient @25°C	80 mA
Process	CHMOS IIE, PX 24

**D.C. CHARACTERISTICS** T<sub>A</sub> = 0°C to 70°C, V<sub>CC</sub> = 5.0V ± 5%

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V <sub>IH</sub> <sup>(5)</sup>	HIGH Level Input Voltage		2.0		V <sub>CC</sub> + 0.3	V
V <sub>IL</sub> <sup>(5)</sup>	LOW Level Input Voltage		-0.3		0.8	V
V <sub>OH</sub> <sup>(6)</sup>	HIGH Level Output Voltage	I <sub>O</sub> = -4.0 mA DC, V <sub>CC</sub> = Min.	2.4			V
V <sub>OL</sub>	LOW Level Output Voltage	I <sub>O</sub> = 4.0 mA DC, V <sub>CC</sub> = Min.			0.45	V
I <sub>I</sub>	Input Leakage Current	V <sub>CC</sub> = Max., GND < V <sub>IN</sub> < V <sub>CC</sub>			±10.0	μA
I <sub>OZ</sub>	Output Leakage Current	V <sub>CC</sub> = Max., GND < V <sub>OUT</sub> < V <sub>CC</sub>			±10.0	μA
I <sub>SC</sub> <sup>(7)</sup>	Output Short Circuit Current	V <sub>CC</sub> = Max., V <sub>OUT</sub> = 0.5V		20	30	mA
I <sub>SB</sub> <sup>(8)</sup>	Standby Current (Standby)	V <sub>CC</sub> = Max., V <sub>IN</sub> = V <sub>CC</sub> or GND		50	100	μA

**D.C. CHARACTERISTICS**  $T_A = 0^\circ\text{C to } 70^\circ\text{C}$ ,  $V_{CC} = 5.0\text{V} \pm 5\%$  (Continued)

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
$I_{CC}$	Power Supply Current (Active) (Turbo Bit Off) Device Prog. as 16-Bit Ctr. (See $I_{CC}$ vs. Freq. Graph.)	$V_{CC} = \text{Max.}$ , $V_{IN} = V_{CC}$ or GND	No Load, Input Freq. = 1 MHz		10	15	mA

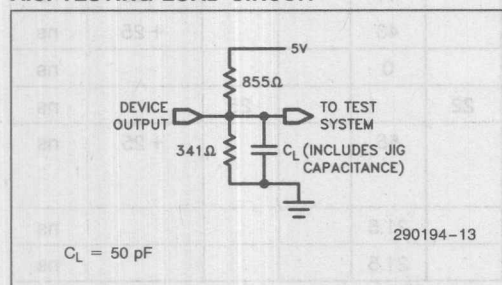
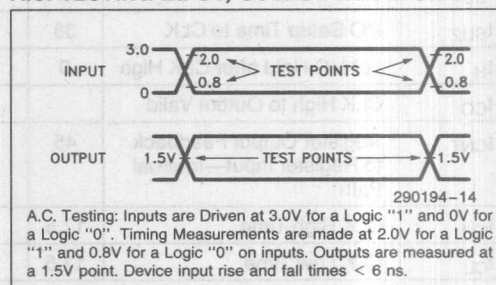
**NOTES:**

5. Absolute values with respect to device GND; all over and undershoots due to system or tester noise are included.

6.  $I_O$  at CMOS levels (3.84V) = -2 mA.

7. Not more than 1 output should be tested at a time. Duration of that test must not exceed 1 second.

8. With Turbo Bit Off, device automatically enters standby mode approximately 100 ns after last input transition.

**A.C. TESTING LOAD CIRCUIT****A.C. TESTING INPUT, OUTPUT WAVEFORM**

2

**CAPACITANCE**

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$C_{IN}$	Input Capacitance	$V_{IN} = 0\text{V}$ , $f = 1.0 \text{ MHz}$			20	pF
$C_{OUT}$	Output Capacitance	$V_{OUT} = 0\text{V}$ , $f = 1.0 \text{ MHz}$			20	pF
$C_{CLK}$	Clock Pin Capacitance	$V_{IN} = 0\text{V}$ , $f = 1.0 \text{ MHz}$			20	pF
$C_{VPP}$	$V_{PP}$ Pin	CLK2 on 5C060, $f = 1.0 \text{ MHz}$			50	pF

**A.C. CHARACTERISTICS**  $T_A = 0^\circ\text{C to } 70^\circ\text{C}$ ,  $V_{CC} = 5\text{V} \pm 5\%$ , Turbo Bit On<sup>(9)</sup>

Symbol	From	To	Device						Non-(11) Turbo Mode	Unit
			5C060-45 EP600-3			5C060-55 EP600				
			Min	Typ	Max	Min	Typ	Max		
t <sub>PD1</sub>	Input	Comb. Output			43			53	+ 25	ns
t <sub>PD2</sub>	I/O	Comb. Output			45			55	+ 25	ns
t <sub>PZX</sub> <sup>(10)</sup>	I or I/O	Output Enable			45			55	+ 25	ns
t <sub>PXZ</sub> <sup>(10)</sup>	I or I/O	Output Disable			45			55	+ 25	ns
t <sub>CLR</sub>	Asynch. Reset	Q Reset			45			55	+ 25	ns

**NOTES:**

9. Typical Values are at  $T_A = 25^\circ\text{C}$ ,  $V_{CC} = 5\text{V}$ , Active Mode.

10.  $t_{PZX}$  and  $t_{PXZ}$  are measured at  $\pm 0.5\text{V}$  from steady state voltage as driven by spec. output load.  $t_{PXZ}$  is measured with  $C_L = 5 \text{ pF}$ .

11. If device is operated with Turbo Bit Off (Non-Turbo Mode), and the device has been inactive for approx. 100 ns, increase time by amount shown.

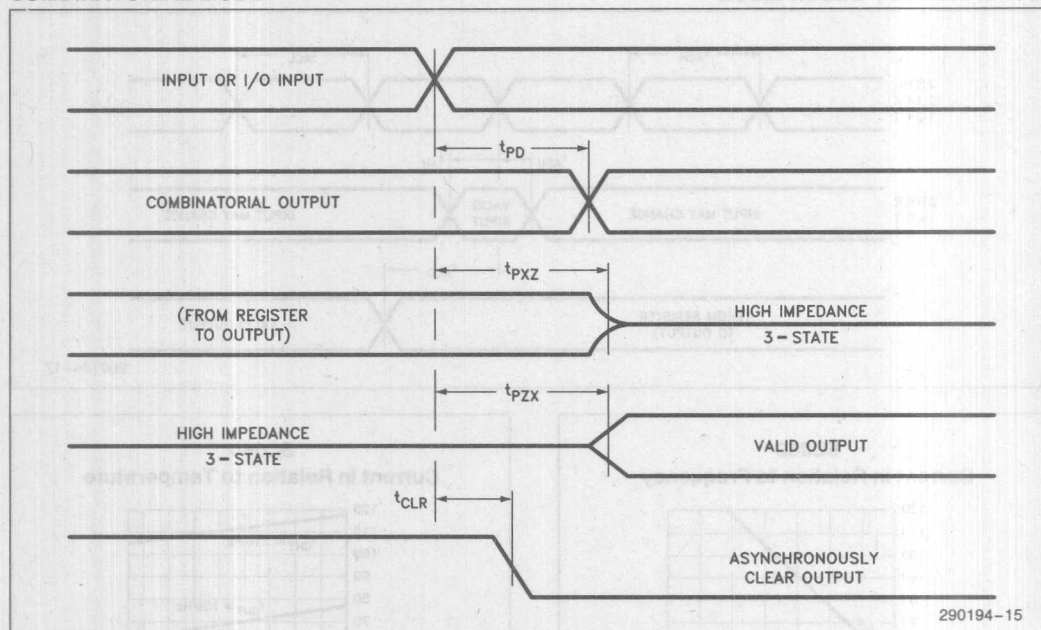
**SYNCHRONOUS CLOCK MODE A.C. CHARACTERISTIC** $T_A = 0^\circ\text{C to } 70^\circ\text{C}$ ,  $V_{CC} = 5.0\text{V} \pm 5\%$ , Turbo Bit On<sup>(9)</sup>

Symbol	Parameter	Device						Non-(11) Turbo Mode	Unit
		5C060-45 EP600-3			5C060-55 EP600				
		Min	Typ	Max	Min	Typ	Max		
f <sub>MAX</sub>	Max. Frequency (Pipelined) (1/t <sub>SU</sub> —No Feedback)			26.3			23.3		MHz
f <sub>CNT</sub>	Max. Count Frequency (1/t <sub>CNT</sub> —With Feedback)			22.2			18.2		MHz
t <sub>SU1</sub>	Input Setup Time to CLK	36			41			+ 25	ns
t <sub>SU2</sub>	I/O Setup Time to CLK	38			43			+ 25	ns
t <sub>H</sub>	I or I/O Hold after CLK High	0			0				ns
t <sub>CO</sub>	CLK High to Output Valid			22			25		ns
t <sub>CNT</sub>	Register Output Feedback to Register Input—Internal Path	45			55			+ 25	ns
t <sub>CH</sub>	CLK High Time	17.5			21.5				ns
t <sub>CL</sub>	CLK Low Time	17.5			21.5				ns

**ASYNCHRONOUS CLOCK MODE A.C. CHARACTERISTICS** $T_A = 0^\circ\text{C to } 70^\circ\text{C}$ ,  $V_{CC} = 5.0\text{V} \pm 5\%$ , Turbo Bit On<sup>(8)</sup>

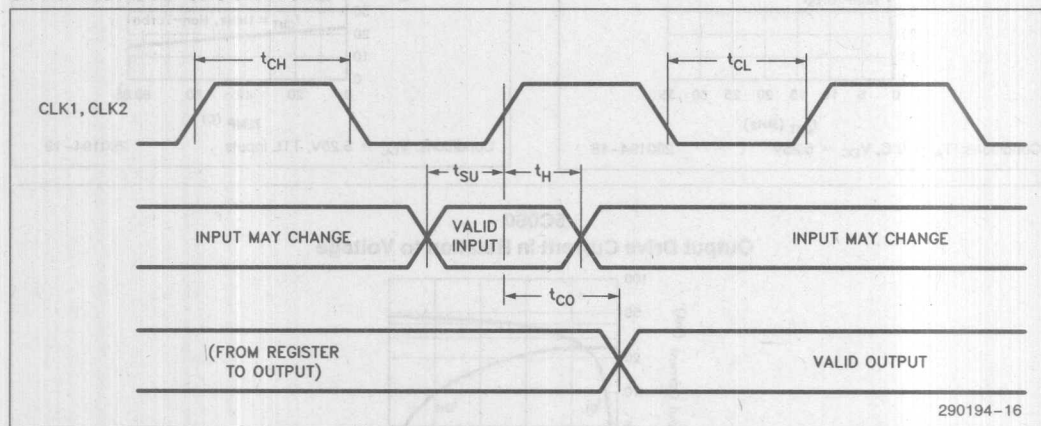
Symbol	Parameter	Device						Non-(11) Turbo Mode	Unit
		5C060-45 EP600-3			5C060-55 EP600				
		Min	Typ	Max	Min	Typ	Max		
f <sub>ACNT</sub>	Max. Count Frequency (1/t <sub>ACNT</sub> —With Feedback)			22.2			18.2		MHz
t <sub>ASU1</sub>	Input Setup Time to Asynch. Clock	10			10			+ 25	ns
t <sub>ASU2</sub>	I/O Setup Time to Asynch. Clock	12			12			+ 25	ns
t <sub>AH</sub>	Input or I/O Hold After Asynch. Clock	15			15				ns
t <sub>ACO</sub>	Asynch. CLK to Output Valid			50			58	+ 25	ns
t <sub>ACNT</sub>	Register Output Feedback to Register Input—Internal Path	45			55			+ 25	ns
t <sub>ACH</sub>	Asynch. CLK High Time	17.5			21.5			+ 25	ns
t <sub>ACL</sub>	Asynch. CLK Low Time	17.5			21.5			+ 25	ns

## COMBINATORIAL MODE



2

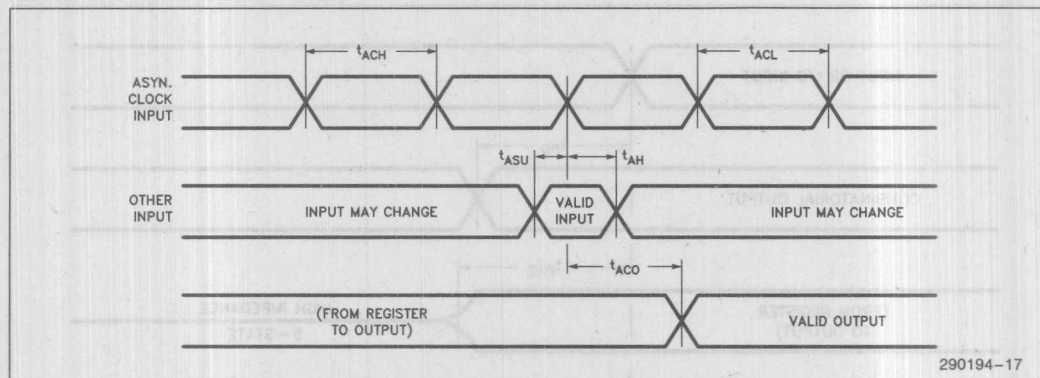
## SYNCHRONOUS CLOCK MODE





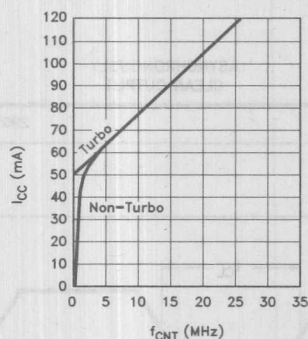
# SWITCHING WAVEFORMS (Continued)

## ASYNCHRONOUS CLOCK MODE



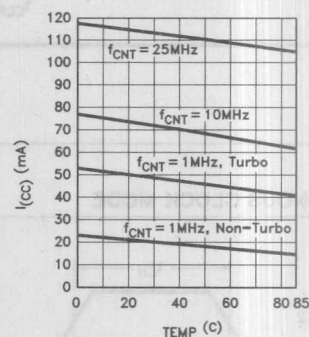
290194-17

**5C060**  
Current in Relation to Frequency



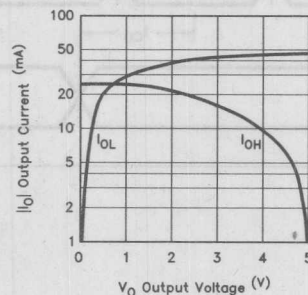
Conditions:  $T_A = 0^\circ\text{C}$ ,  $V_{CC} = 5.25\text{V}$  290194-18

**5C060**  
Current in Relation to Temperature



Conditions:  $V_{CC} = 5.25\text{V}$ , TTL inputs 290194-19

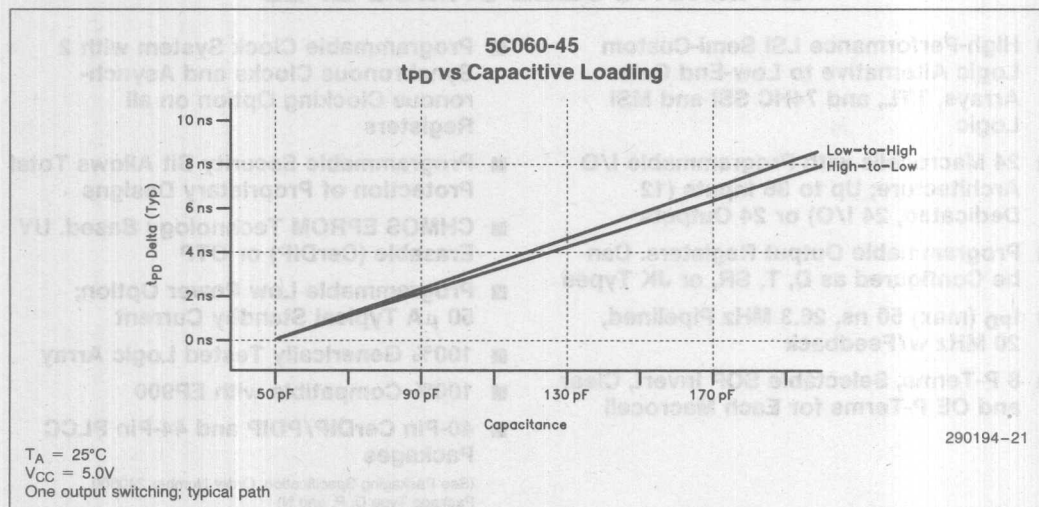
**5C060**  
Output Drive Current in Relation to Voltage



Conditions:  $T_A = 25^\circ\text{C}$

290194-20

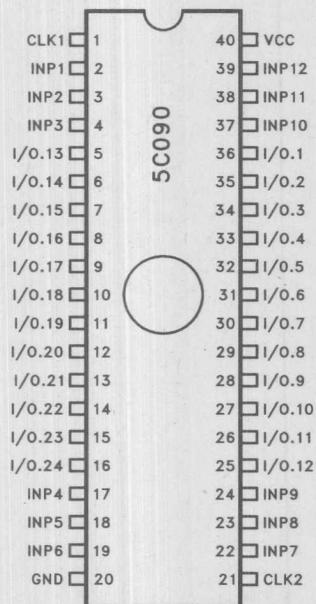
## SWITCHING WAVEFORMS (Continued)



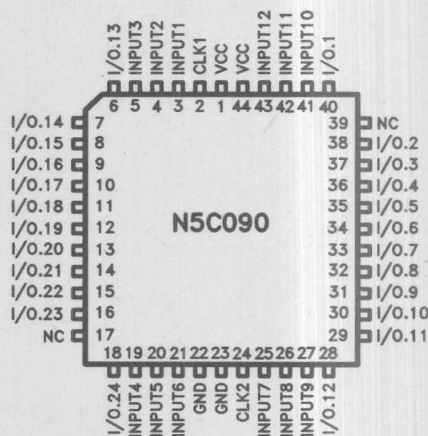
# 24-MACROCELL CHMOS EPLD

- High-Performance LSI Semi-Custom Logic Alternative to Low-End Gate Arrays, TTL, and 74HC SSI and MSI Logic
- 24 Macrocells with Programmable I/O Architecture; Up to 36 Inputs (12 Dedicated, 24 I/O) or 24 Outputs
- Programmable Output Registers. Can be Configured as D, T, SR, or JK Types
- $t_{PD}$  (max) 50 ns, 26.3 MHz Pipelined, 20 MHz w/Feedback
- 8 P-Terms, Selectable SOP Invert, Clear and OE P-Terms for Each Macrocell
- Programmable Clock System with 2 Synchronous Clocks and Asynchronous Clocking Option on all Registers
- Programmable Security Bit Allows Total Protection of Proprietary Designs
- CHMOS EPROM Technology Based. UV Erasable (CerDIP) or OTP
- Programmable Low Power Option; 50  $\mu$ A Typical Standby Current
- 100% Generically Tested Logic Array
- 100% Compatible with EP900
- 40-Pin CerDIP/PDIP and 44-Pin PLCC Packages

(See Packaging Specification, Order Number 240800, Package Type D, P, and N)



290195-1



290195-2

Figure 1. 5C090 Pin Configurations

The Intel 5C090 EPLD (Erasable Programmable Logic Device) is a 24-macrocell, 40-pin, general-purpose device. The device can be used to replace low-end gate arrays, multiple programmable logic arrays and LS TTL and 74HC (CMOS) SSI and MSI logic devices. With its revolutionary programmable I/O architecture, the device has advanced functional capabilities beyond that of typical programmable logic. Figure 2 shows the global architecture of the device.

The 5C090 EPLD uses CHMOS EPROM (floating gate) cells as logic control elements instead of fuses. The CHMOS EPROM technology reduces power consumption of EPLDs to less than 20% of a comparable bipolar device without sacrificing speed performance. In addition, Intel's advanced CHMOS II-E EPROM process technology enables greater logic densities to be achieved with superior speed and low-power performance over other comparable devices. Intel's EPLDs add the benefits of "zero" stand-by power not available on other programmable logic devices. EPROM technology allows these devices to be 100% factory tested by programming and erasing all the EPROM logic control elements.

The architecture of the 5C090 is based on the "Sum of Products" PLA (Programmable Logic Array) structure with a programmable AND array feeding into a fixed OR array. The device accommodates combinational and sequential logic functions. A proprietary programmable I/O architecture provides individual selection of either combinational or registered output and feedback signals all with selectable polarity.

A feature unique to the 5C090 is the ability to individually program the output registers as a D-, T-, SR-, or JK-type Flip-Flop without sacrificing the utilization of programmable AND logic. Additionally, each output register can be individually clocked from any of the input or feedback paths available within the AND array. With these features, a wide variety of logic functions can be simultaneously implemented—all on the same device.

## ARCHITECTURE DESCRIPTION

The 5C090 has 12 dedicated inputs, 24 I/O pins which may be configured for input, output, or bidirectional operations, and 2 synchronous clock inputs. The 5C090 is packaged in a 40-lead windowed ceramic DIP or 44-lead plastic leaded chip carrier package and contains 24 programmable registers.

2

The basic Macrocell architecture for the 5C090 is shown in Figure 3. The 5C090 has 24 of these macrocells (one for each I/O pin). The Macrocell is organized in the familiar sum-of-products structure with a programmable AND array attached to a fixed OR term. The inputs to the programmable AND array originate from the true and complement signals from each of the dedicated input pins and each of the I/O control blocks.

The AND array for the 5C090 has 72 inputs derived from the true and complement signals at the input and I/O pins. The AND array in the 5C090 encompasses 240 product terms which are distributed among the 24 Macrocells.

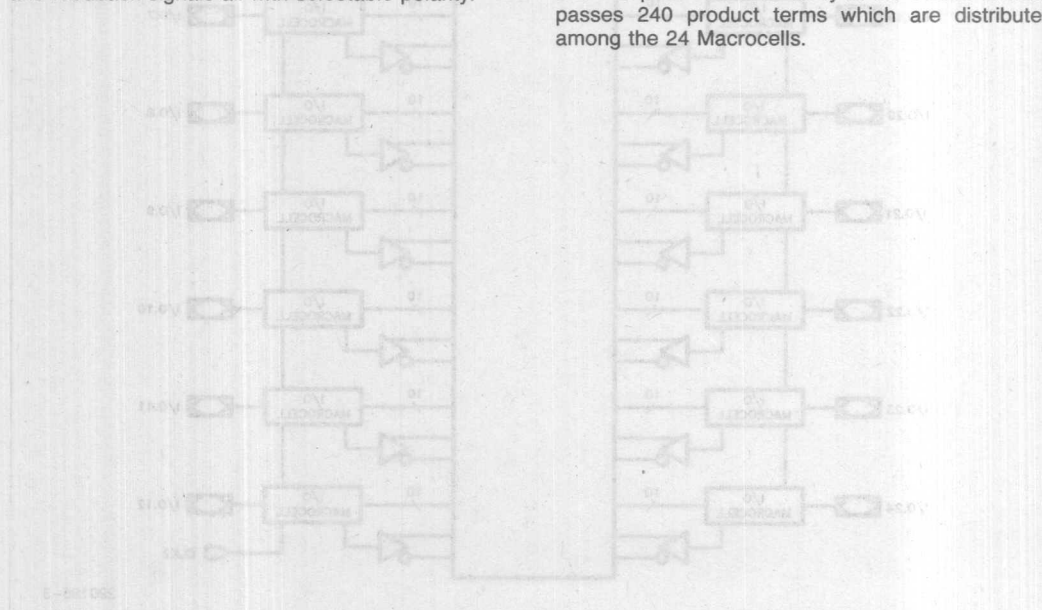


Figure 2. 5C090 Global Architecture



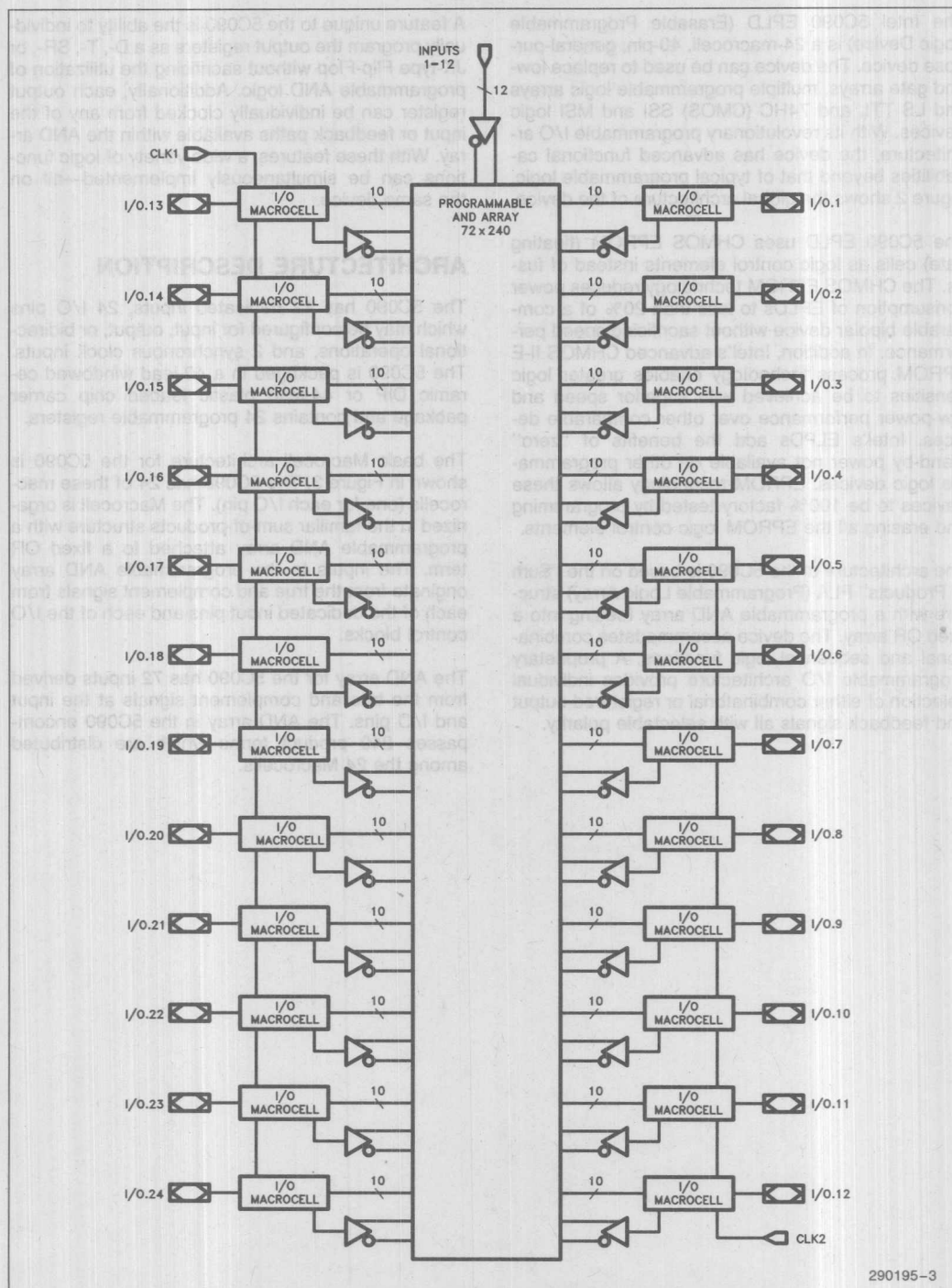


Figure 2. 5C090 Global Architecture

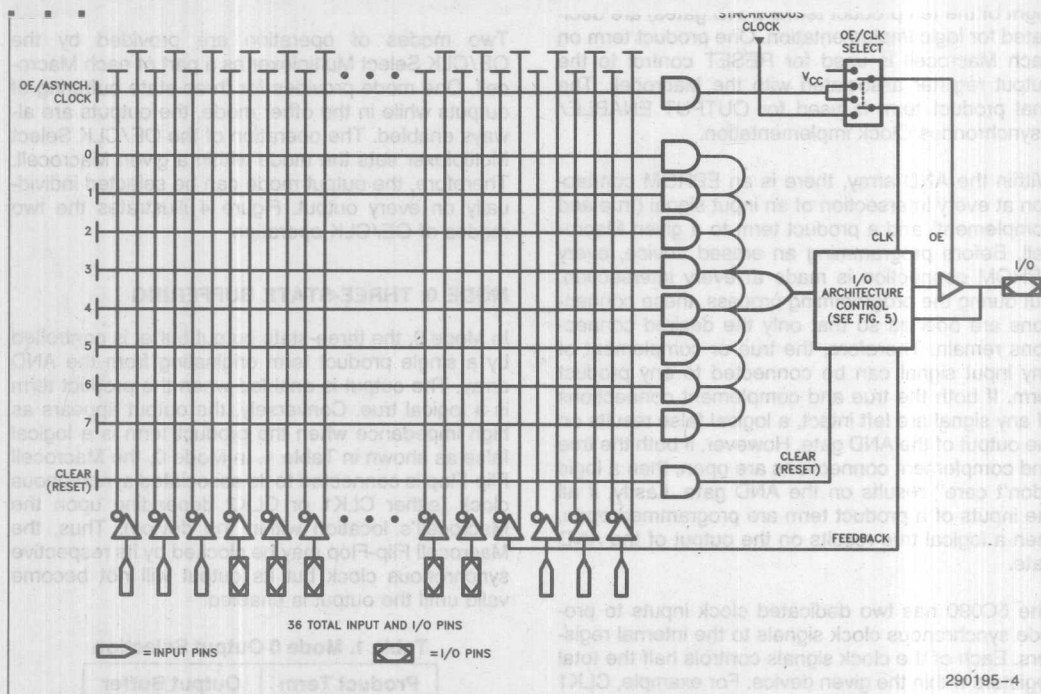


Figure 3. 5C090 Macrocell Architecture

2

The Macrocells contain ten product terms total. Eight of the ten product terms (AND gates) are dedicated for logic implementation. One product term on each Macrocell is used for RESET control to the output register associated with the Macrocell. The final product term is used for OUTPUT ENABLE/Asynchronous Clock implementation.

Within the AND array, there is an EPROM connection at every intersection of an input signal (true and complement) and a product term to a given Macrocell. Before programming an erased device, every EPROM connection is made at every intersection. But during the programming process, these connections are opened so that only the desired connections remain. Therefore, the true or complement of any input signal can be connected to any product term. If both the true and complement connections of any signal are left intact, a logical false results on the output of the AND gate. However, if both the true and complement connections are open, then a logic "don't care" results on the AND gate. Lastly, if all the inputs of a product term are programmed open, then a logical true results on the output of the AND gate.

The 5C090 has two dedicated clock inputs to provide synchronous clock signals to the internal registers. Each of the clock signals controls half the total registers within the given device. For example, CLK1 provides synchronous clocking to the registers in Macrocells in the left half of the array while CLK2 controls the registers associated with Macrocells in the right half of the array. The advanced I/O architecture allows for any number of the registers to be synchronously clocked (from none to all). Both of the dedicated clock inputs latch the data into a given register when triggered on a positive edge.

## MACROCELL ARCHITECTURE SELECTION

The 5C090 architecture provides each Macrocell with over 50 different possible I/O register configurations. Each I/O pin can be configured for combinatorial or registered output (true or complement) with feedback. In addition, four different types of output registers can be implemented into every I/O pin without any additional logic requirements. The feedback mechanism for each register back into the AND array can be programmed to provide for either registered feedback from the Macrocell or input feedback (treating the pin as an input). Another advantage of the advanced I/O capability of the 5C090 is the ability to individually clock each internal register from asynchronous clock signals.

## Output Enable (OE)/Clock Selection

Two modes of operation are provided by the OE/CLK Select Multiplexer as a part of each Macrocell. One mode provides for three-state buffering of outputs while in the other mode, the outputs are always enabled. The operation of the OE/CLK Select Multiplexer sets the mode within a given Macrocell. Therefore, the output mode can be selected individually on every output. Figure 4 illustrates the two modes of OE/CLK operation.

### MODE 0: THREE-STATE BUFFERING

In Mode 0, the three-state output buffer is controlled by a single product term originating from the AND array. The output is enabled when the product term is a logical true. Conversely, the output appears as high impedance when the product term is a logical false as shown in Table 1. In Mode 0, the Macrocell Flip-Flop is connected to its associated synchronous clock (either CLK1 or CLK2 depending upon the Macrocell's location within the device). Thus, the Macrocell Flip-Flop may be clocked by its respective synchronous clock but its output will not become valid until the output is enabled.

Table 1. Mode 0 Output Selection

Product Term	Output Buffer
FALSE	Three-State
TRUE	Enabled

### MODE 1: OUTPUT BUFFER ENABLED

In Mode 1, the Output Buffer is always enabled. In addition, the Macrocell Flip-Flop is connected to the AND array. The Macrocell Flip-Flop may now be triggered from an asynchronous clock signal generated by the AND array logic to the OE/CLK multiplexable term. Mode 1 allows the Macrocell Flip-Flops to be individually clocked from any of the available signals in the AND array. Since both true and complement values appear in the AND array, the Flip-Flop may be clocked by positive-or negative-going signals at any input pin. Gated clock structures can be created since the Flip-Flop clock is created by a product term.

## Invert Select EPROM Bit

The Invert Select EPROM bit is used to invert the product term input into the register. This applies to all inputs including double inputs on the JK and SR registers.

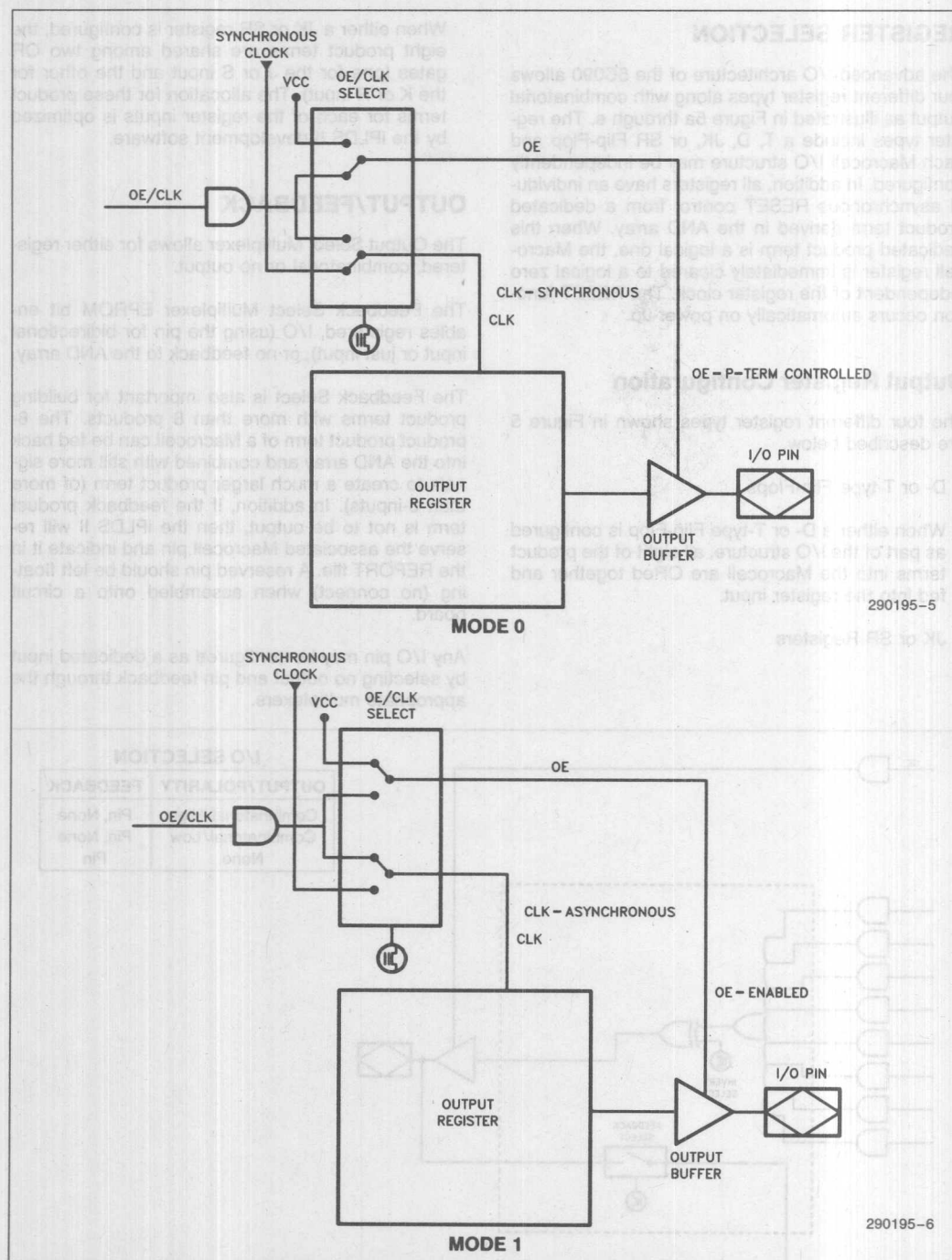


Figure 4. Output Enable/Clock Configuration



## REGISTER SELECTION

The advanced I/O architecture of the 5C090 allows four different register types along with combinatorial output as illustrated in Figure 5a through e. The register types include a T, D, JK, or SR Flip-Flop and each Macrocell I/O structure may be independently configured. In addition, all registers have an individual asynchronous RESET control from a dedicated product term derived in the AND array. When this dedicated product term is a logical one, the Macrocell register is immediately cleared to a logical zero independent of the register clock. The RESET function occurs automatically on power-up.

## Output Register Configuration

The four different register types shown in Figure 5 are described below.

### D- or T-type Flip-Flops

When either a D- or T-type Flip-Flop is configured as part of the I/O structure, all eight of the product terms into the Macrocell are ORed together and fed into the register input.

### JK or SR Registers

When either a JK or SR register is configured, the eight product terms are shared among two OR gates (one for the J or S input and the other for the K or R input). The allocation for these product terms for each of the register inputs is optimized by the iPLDS II development software.

## OUTPUT/FEEDBACK

The Output Select Multiplexer allows for either registered, combinatorial or no output.

The Feedback Select Multiplexer EPROM bit enables registered, I/O (using the pin for bidirectional input or just input), or no feedback to the AND array.

The Feedback Select is also important for building product terms with more than 8 products. The 8-product product term of a Macrocell can be fed back into the AND array and combined with still more signals to create a much larger product term (of more than 8-inputs). In addition, if the feedback product term is not to be output, then the iPLDS II will reserve the associated Macrocell pin and indicate it in the REPORT file. A reserved pin should be left floating (no connect) when assembled onto a circuit board.

Any I/O pin may be configured as a dedicated input by selecting no output and pin feedback through the appropriate multiplexers.

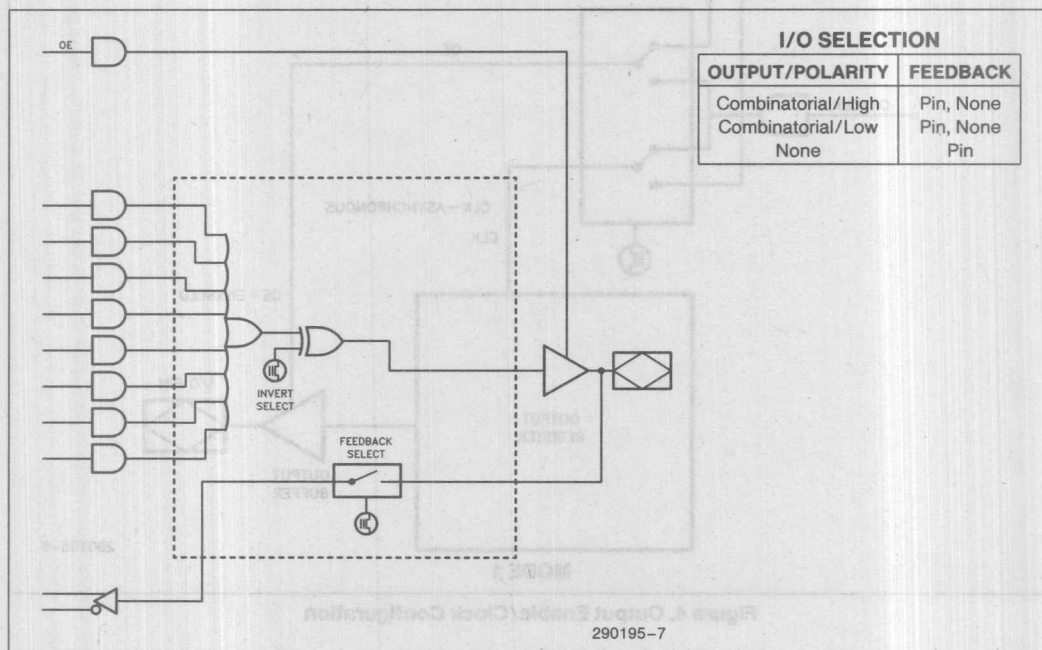


Figure 5a. Combinatorial I/O Configuration

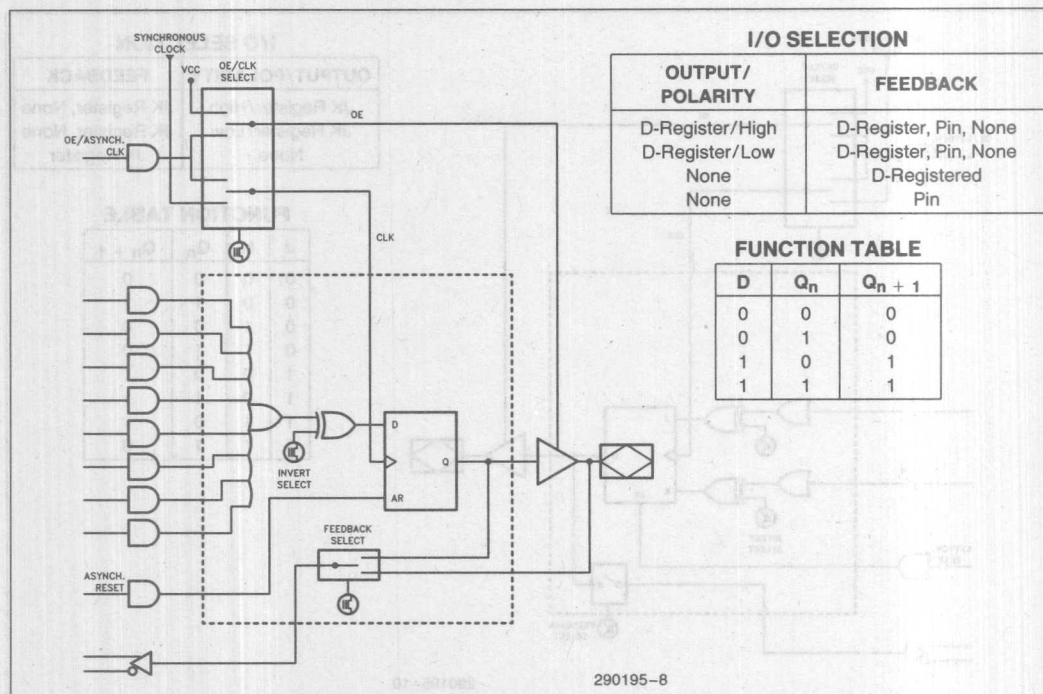


Figure 5b. D-Type Flip-Flop Register Configuration

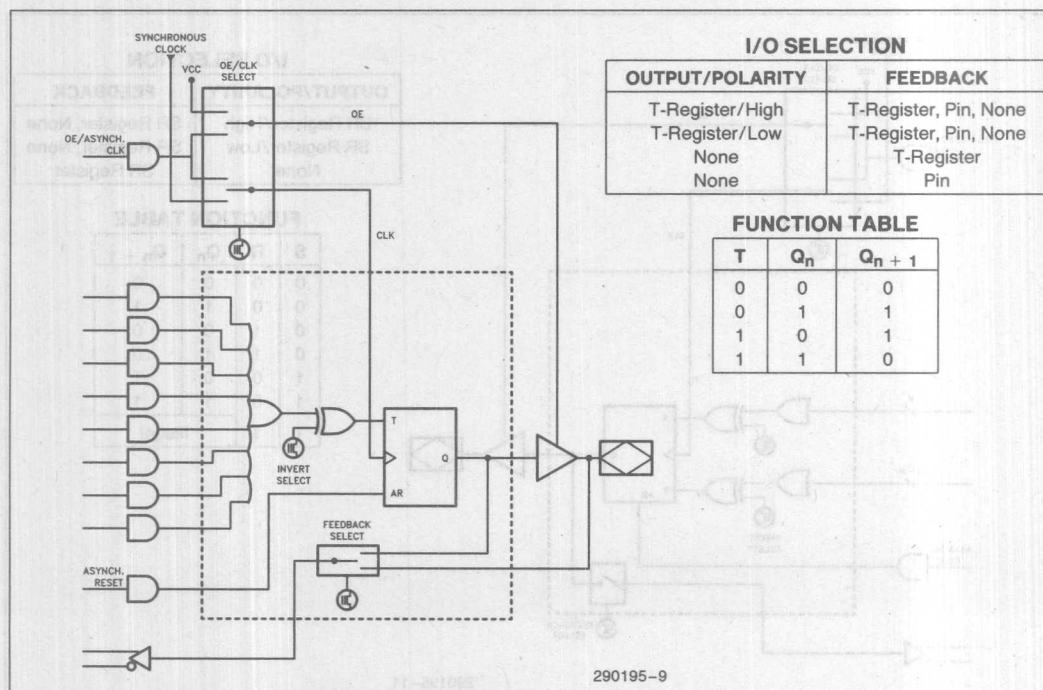
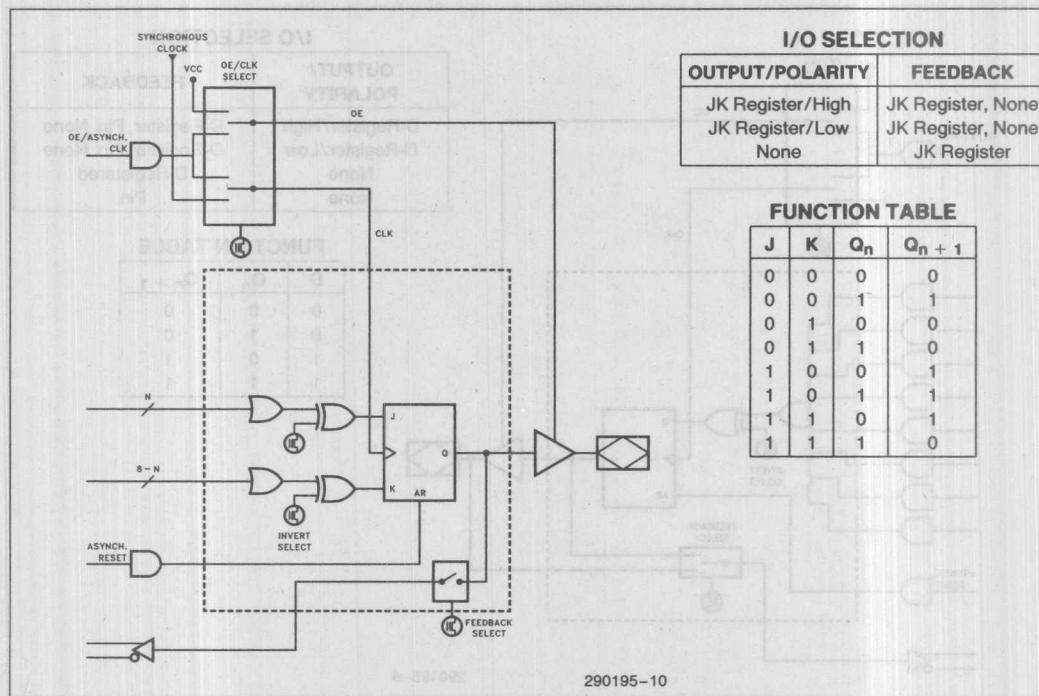
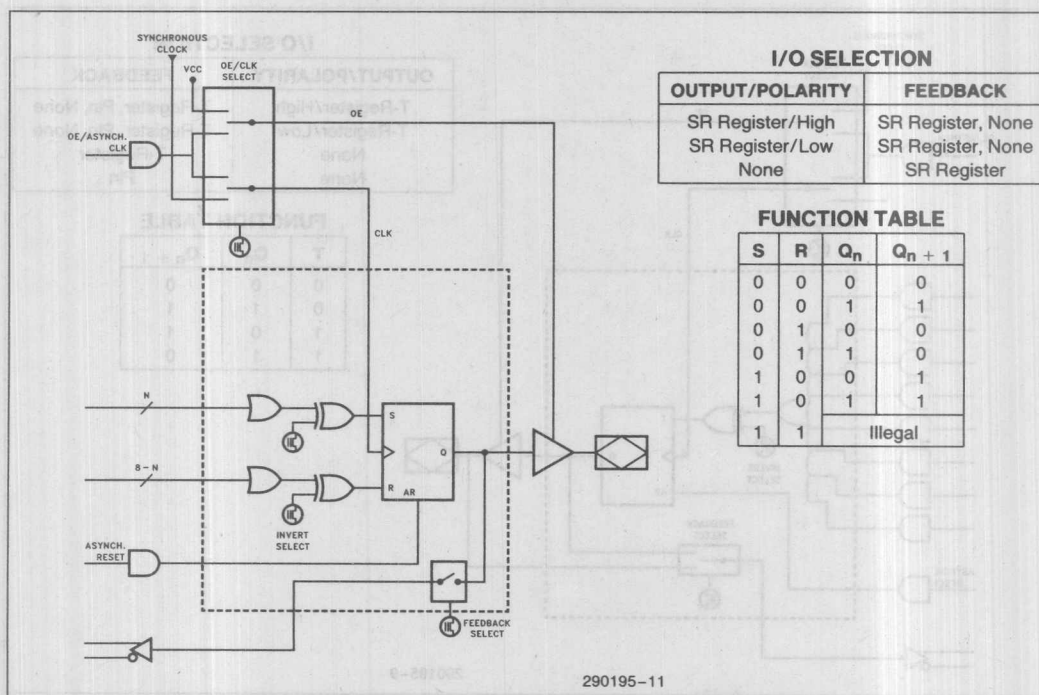


Figure 5c. Toggle Flip-Flop Register Configuration



### Figure 5d. JK Flip-Flop Register Configuration



### Figure 5e. SR Flip-Flop Register Configuration

Prior to programming or after erasing, the I/O structure is configured for combinatorial active low output with input (pin) feedback.

## ERASURE CHARACTERISTICS

Erasure characteristics of the device are such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000Å. It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000–4000Å. Data shows that constant exposure to room level fluorescent lighting could erase the typical device in approximately three years, while it would take approximately one week to cause erasure when exposed to direct sunlight. If the 5C090 is to be exposed to these types of lighting conditions for extended periods of time, conductive opaque labels should be placed over the device window to prevent unintentional erasure.

The recommended erasure procedure for the 5C090 is exposure to shortwave ultraviolet light with a wavelength of 2537Å. The integrated dose (i.e., UV intensity  $\times$  exposure time) for erasure should be a minimum of fifteen (15) Wsec/cm<sup>2</sup>. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12,000  $\mu$ W/cm<sup>2</sup> power rating. The 5C090 should be placed within one inch of the lamp tubes during erasure. The maximum integrated dose the 5C090 can be exposed to without damage is 7258 Wsec/cm<sup>2</sup> (1 week at 12,000  $\mu$ W/cm<sup>2</sup>). Exposure to high intensity UV light for longer periods may cause permanent damage to the device.

## PROGRAMMING CHARACTERISTICS

Initially, and after erasure, all the EPROM control bits of the 5C090 are connected (in the "1" state). Each of the connected control bits are selectively disconnected by programming the EPROM cells into their "0" state. Programming voltage and waveform specifications are available by request from Intel to support programming of the 5C090.

### Intelligent Programming Algorithm

The 5C090 supports the Intelligent Programming Algorithm which rapidly programs Intel ELPDs using an efficient and reliable method. The Intelligent Programming Algorithm is particularly suited to the production programming environment. This method

of each bit is continually monitored to determine when the bit has been successfully programmed.

## FUNCTIONAL TESTING

Since the logical operation of the 5C090 is controlled by EPROM elements, the device is completely testable. Each programmable EPROM bit controlling the internal logic is tested using application-independent test program patterns. After testing, the devices are erased before shipment to customers. No post-programming tests of the EPROM array are required.

The testability and reliability of EPROM-based programmable logic devices is an important feature over similar devices based on fuse technology. Fuse-based programmable logic devices require a user to perform post-programming tests to insure proper programming. These tests must be done at the device level because of the cumulative error effect. For example, a board containing ten devices each possessing a 2% device fallout translates into an 18% fallout at the board level (it should be noted that programming fallout of fuse-based programmable logic devices is typically 2% or higher).

## DESIGN RECOMMENDATIONS

For proper operation, it is recommended that all input and output pins be constrained to the voltage range  $GND < (V_{IN} \text{ or } V_{OUT}) < V_{CC}$ . Unused inputs and I/Os should be tied to  $V_{CC}$  or  $GND$  to minimize device power consumption. Reserved pins (as indicated in the logic compiler REPORT file) should be left floating (no connect) so that the pin can attain the appropriate logic level. A power supply decoupling capacitor of at least 0.2  $\mu$ F must be connected directly between  $V_{CC}$  and  $GND$  pins of the device.

As with all CMOS devices, ESD handling procedures should be used with the 5C090 to prevent damage to the device during programming, assembly and test.

## DESIGN SECURITY

A single EPROM bit provides a programmable design security feature that controls the access to the data programmed into the device. If this bit is set, a proprietary design within the device cannot be copied. This EPROM security bit enables a higher de-



gree of design security than fused-based devices since programmed data within EPROM cells is invisible even to microscopic evaluation. The EPROM security bit, along with all the other EPROM control bits, will be reset by erasing the device.

## AUTOMATIC STAND-BY MODE

The 5C090 contains a programmable bit, the Turbo Bit, that optimizes operation for speed or for power savings. When the Turbo Bit is programmed (TURBO = ON), the device is optimized for maximum speed. When the Turbo Bit is not programmed (TURBO = OFF), the device is optimized for power savings by entering standby mode during periods of inactivity.

Figure 6 shows the device entering standby mode approximately 100 ns after the last input transition. When the next input transition is detected, the device returns to active mode. Wakeup time adds an additional 25 ns to the propagation delay through the device as measured from the first input. No delay will occur if an output is dependent on more than one input and the last of the inputs changes after the device has returned to active mode.

After erasure, the Turbo Bit is unprogrammed (OFF); automatic standby mode is enabled. When the Turbo Bit is programmed (ON), the device never enters standby mode.

## LATCH-UP IMMUNITY

All of the input, I/O, and clock pins of the 5C090 have been designed to resist latch-up which is inher-

ent in inferior CMOS structures. The 5C090 is designed with Intel's proprietary CHMOS II-E EPROM process. Thus, each of the pins will not experience latch-up with currents up to  $\pm 100$  mA and voltages ranging from  $-1V$  to  $(V_{CC} + 1V)$ . Furthermore, the programming pin is designed to resist latch-up to the 13.5V maximum device limit.

## SOFTWARE SUPPORT

Full logic compilation and functional simulation for the 5C090 is supported by PLDshell Plus™ software. The GUPI Logic-11D provides programming support on Intel programmers.

PLDshell Plus design software is Intel's new, user-friendly design tool for  $\mu$ PLD design. PLDshell Plus allows users to incorporate their preferred text editor, programming software, and additional design tools into an easy-to-use, menued design environment that includes Intel's PLDasm™ logic compiler and simulation software along with disassembly, conversion, and translation utilities. The PLDasm compiler and simulator software accepts industry-standard PDS source files that express designs as Boolean equations, truth tables, or state machines. On-line help, data sheet briefs, technical notes, and error message information, along with waveform viewing/printing capability make the design task as easy as possible. PLDshell Plus software is available from Intel Literature channels or from your local Intel sales representative.

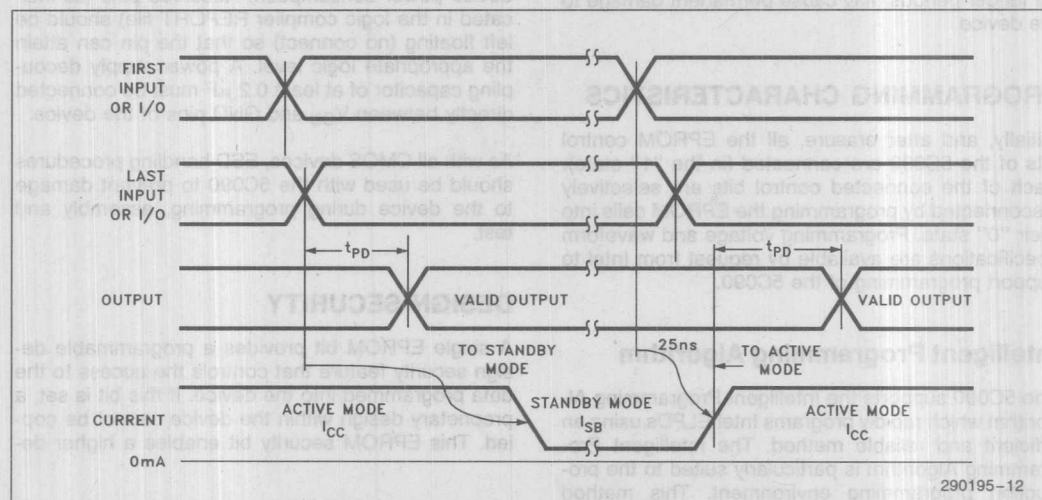


Figure 6. 5C090 Standby and Active Mode Transitions

\*PLDshell Plus™ is a trademark of Intel Corporation.

Tools that support schematic capture and timing simulation for the 5C060 are available. Support under iPLS II is still available. Please refer to the "Development Tools" section of the Programmable Logic Handbook.

The 5C090 is also supported by third-party logic compilers such as ABEL\*, CUPL\*, PLDesigner, Log/IC\*, etc. Programming support is provided by third-party programmer companies such as Data I/O, Logical Devices, STAG, etc. Please refer to the "Third-Party Support" lists in the *Programmable Logic* handbook for complete information and vendor contacts.

$t_p$	Input Fall Time	500 ns
$t_r$	Input Rise Time	500 ns
$T_A$	Operating Temperature	0 to 70 °C
$V_O$	Output Voltage	0 VCC V
$V_{IH}$	Input Voltage	0 VCC V
$V_{CC}$	Supply Voltage	4.5V to 5.5V V

NOTE:

\* At  $t_p$  for CLK is 250 ns max.

## PACKAGE TECHNOLOGY SPECIFICATIONS

Specification	Description
38°C/W—CERDIP	6A—Junction-to-Ambient Thermal Resistance
45°C/W—PDIP	
48°C/W—PLCC	
18°C/W—CERDIP	10C—Junction-to-Case Thermal Resistance
18°C/W—PDIP	
18°C/W—PLCC	
80 mA	10C Hot—Ambient 87°C
80 mA	10C Typical—Ambient 85°C
CMOS like P <sub>DISS</sub>	Power Dissipation

## D.C. CHARACTERISTICS $T_A = 0^\circ\text{C to } 70^\circ\text{C}, V_{CC} = 5.0\text{V} \pm 5\%$

Symbol	Parameter	Conditions	Min	Typ	Max
$V_{IH}$ (1)	HIGH Level Input Voltage		2.0		$V_{CC} - 0.3$ V
$V_{IL}$ (2)	LOW Level Input Voltage		-0.3		0.8 V
$V_{OH}$ (3)	HIGH Level Output Voltage	$I_O = -40$ mA DC, $V_{CC} = \text{Min}$	2.4		V
$V_{OL}$ (4)	LOW Level Output Voltage	$I_O = 40$ mA DC, $V_{CC} = \text{Min}$			0.45 V
$I_{IH}$ (5)	Input Leakage Current	$V_{CC} = \text{Max}, \text{GND} < V_{IH} < V_{CC}$		1.00	$\mu\text{A}$
$I_{OL}$ (6)	Output Leakage Current	$V_{CC} = \text{Max}, \text{GND} < V_{OL} < V_{CC}$		1.00	$\mu\text{A}$
$I_{CCH}$ (7)	CMOS Short Circuit Current	$V_{CC} = \text{Max}, V_{OUT} = 0.5\text{V}$	50		mA
$I_{DD}$ (8)	Supply Current	$V_{CC} = \text{Max}, V_{IH} = V_{CC}$ or GND	100		$\mu\text{A}$

\*ABEL is a trademark of Data I/O, Corporation. CUPL is a trademark of Logical Devices, Inc. PLDesigner is a trademark of MINC, Inc. Log/IC is trademark of ISDATA, Inc.

**ABSOLUTE MAXIMUM RATINGS\***

Symbol	Parameter	Min	Max	Units
V <sub>CC</sub>	Supply Voltage <sup>(1)</sup>	-2.0	7.0	V
V <sub>PP</sub>	Programming Supply Voltage <sup>(1)</sup>	-2.0	13.5	V
V <sub>I</sub>	DC Input Voltage <sup>(1)(2)</sup>	-0.5	V <sub>CC</sub> + 0.5	V
t <sub>stg</sub>	Storage Temperature	-65	+150	°C
t <sub>amb</sub>	Ambient Temperature <sup>(3)</sup>	-10	+85	°C

**NOTES:**

1. Voltages with respect to ground.
2. Minimum DC input is -0.5V. During transitions, the inputs may undershoot to -2.0V or overshoot to +7.0V for periods less than 20 ns under no load conditions.
3. Under bias. Extended temperature versions are also available.

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**\*WARNING:** Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.

**RECOMMENDED OPERATING CONDITIONS**

Symbol	Parameter	Min	Max	Unit
V <sub>CC</sub>	Supply Voltage	4.75	5.25	V
V <sub>IN</sub>	Input Voltage	0	V <sub>CC</sub>	V
V <sub>O</sub>	Output Voltage	0	V <sub>CC</sub>	V
T <sub>A</sub>	Operating Temperature	0	+70	°C
t <sub>R</sub>	Input Rise Time		500	ns
t <sub>F</sub>	Input Fall Time		500	ns

**NOTE:**

4. t<sub>R</sub>, t<sub>F</sub> for CLK is 250 ns max.

**PACKAGE/TECHNOLOGY SPECIFICATIONS**

Description	Specification
θ <sub>JA</sub> —Junction-to-Ambient Thermal Resistance	36°C/W—CerDIP 48°C/W—PDIP 48°C/W—PLCC
θ <sub>JC</sub> —Junction-to-Case Thermal Resistance	13°C/W—CerDIP 16°C/W—PDIP 16°C/W—PLCC
I <sub>CC</sub> Hot—Ambient @70°C	90 mA
I <sub>CC</sub> Typical—Ambient @25°C	90 mA
Process	CHMOS IIE, PX 24

**D.C. CHARACTERISTICS** T<sub>A</sub> = 0°C to 70°C, V<sub>CC</sub> = 5.0V ±5%

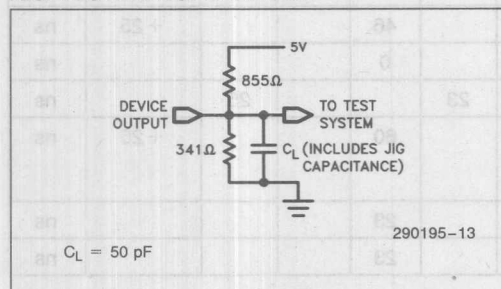
Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V <sub>IH</sub> <sup>(5)</sup>	HIGH Level Input Voltage		2.0		V <sub>CC</sub> + 0.3	V
V <sub>IL</sub> <sup>(5)</sup>	LOW Level Input Voltage		-0.3		0.8	V
V <sub>OH</sub> <sup>(6)</sup>	HIGH Level Output Voltage	I <sub>O</sub> = -4.0 mA DC, V <sub>CC</sub> = Min.	2.4			V
V <sub>OL</sub>	LOW Level Output Voltage	I <sub>O</sub> = 4.0 mA DC, V <sub>CC</sub> = Min.			0.45	V
I <sub>I</sub>	Input Leakage Current	V <sub>CC</sub> = Max., GND < V <sub>IN</sub> < V <sub>CC</sub>			±10.0	μA
I <sub>OZ</sub>	Output Leakage Current	V <sub>CC</sub> = Max., GND < V <sub>OUT</sub> < V <sub>CC</sub>			±10.0	μA
I <sub>SC</sub> <sup>(7)</sup>	Output Short Circuit Current	V <sub>CC</sub> = Max., V <sub>OUT</sub> = 0.5V		20	30	mA
I <sub>SB</sub> <sup>(8)</sup>	Standby Current (Standby)	V <sub>CC</sub> = Max., V <sub>IN</sub> = V <sub>CC</sub> or GND		50	150	μA

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
$I_{CC}$	Power Supply Current (Active) (Turbo Bit Off) Device Prog. as Two 12-Bit Ctrs. (See $I_{CC}$ vs. Freq. Graph)	$V_{CC} = \text{Max.}$ , $V_{IN} = V_{CC}$ or GND	No Load, Input Freq. = 1 MHz		15	25	mA

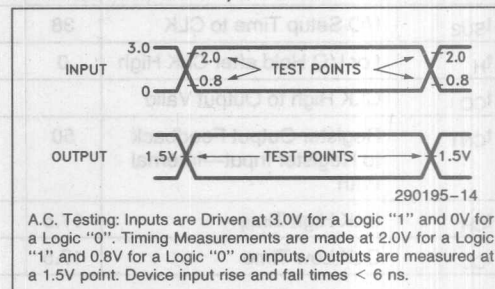
#### NOTES:

5. Absolute values with respect to device GND; all over and undershoots due to system or tester noise are included.
6.  $I_O$  at CMOS levels (3.84V) = -2 mA.
7. Not more than 1 output should be tested at a time. Duration of that test must not exceed 1 second.
8. With Turbo Bit Off, device automatically enters standby mode approximately 100 ns after last input transition.

#### A.C. TESTING LOAD CIRCUIT



#### A.C. TESTING INPUT, OUTPUT WAVEFORM



2

#### CAPACITANCE

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$C_{IN}$	Input Capacitance	$V_{IN} = 0V$ , $f = 1.0 \text{ MHz}$			20	pF
$C_{OUT}$	Output Capacitance	$V_{OUT} = 0V$ , $f = 1.0 \text{ MHz}$			20	pF
$C_{CLK}$	Clock Pin Capacitance	$V_{IN} = 0V$ , $f = 1.0 \text{ MHz}$			20	pF
$C_{VPP}$	$V_{PP}$ Pin	CLK2 on 5C090, $f = 1.0 \text{ MHz}$			80	pF

#### A.C. CHARACTERISTICS $T_A = 0^\circ\text{C}$ to $70^\circ\text{C}$ , $V_{CC} = 5V \pm 5\%$ , Turbo Bit On<sup>(9)</sup>

Symbol	From	To	Device						Non-(11) Turbo Mode	Unit
			5C090-50 EP900-2			5C090-60 EP900				
			Min	Typ	Max	Min	Typ	Max		
t <sub>PD1</sub>	Input	Comb. Output			45			55	+ 25	ns
t <sub>PD2</sub>	I/O	Comb. Output			50			60	+ 25	ns
t <sub>PZX</sub> <sup>(10)</sup>	I or I/O	Output Enable			50			60	+ 25	ns
t <sub>PXZ</sub> <sup>(10)</sup>	I or I/O	Output Disable			50			60	+ 25	ns
t <sub>CLR</sub>	Asynch. Reset	Q Reset			50			60	+ 25	ns

#### NOTES:

9. Typical Values are at  $T_A = 25^\circ\text{C}$ ,  $V_{CC} = 5V$ , Active Mode.
10.  $t_{PZX}$  and  $t_{PXZ}$  are measured at  $\pm 0.5V$  from steady state voltage as driven by spec. output load.  $t_{PXZ}$  is measured with  $C_L = 5 \text{ pF}$ .
11. If device is operated with Turbo Bit Off (Non-Turbo Mode) and the device has been inactive for approx. 100 ns, increase time by amount shown.



**SYNCHRONOUS CLOCK MODE A.C. CHARACTERISTIC** $T_A = 0^\circ\text{C to } 70^\circ\text{C}$ ,  $V_{CC} = 5.0\text{V} \pm 5\%$ , Turbo Bit On<sup>(9)</sup>

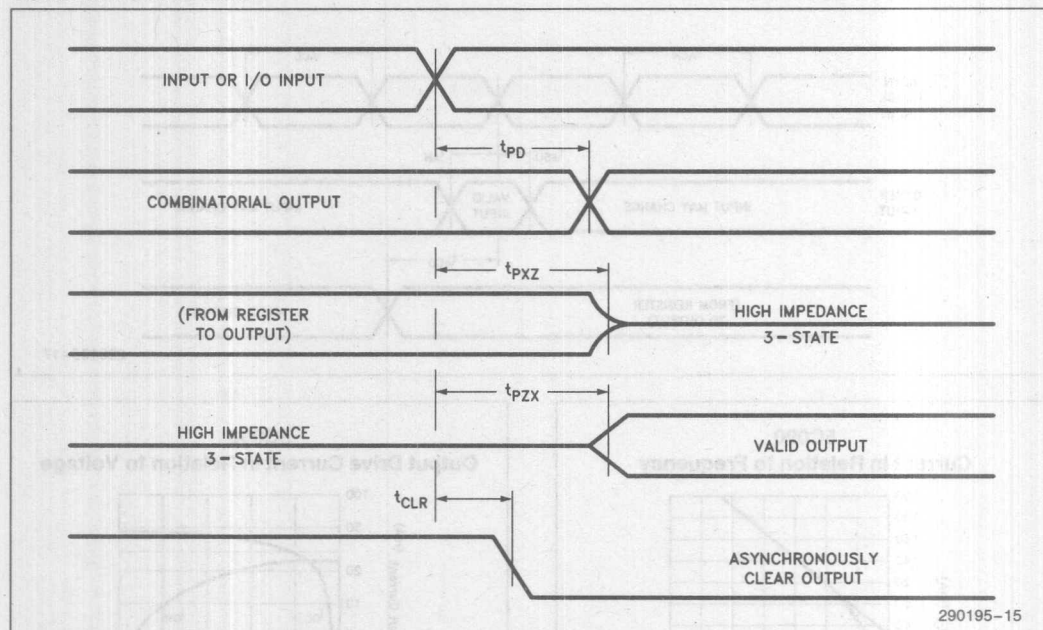
Symbol	Parameter	Device						Non-(11) Turbo Mode	Unit
		5C090-50 EP900-2			5C090-60 EP900				
		Min	Typ	Max	Min	Typ	Max		
f <sub>MAX</sub>	Max. Frequency (Pipelined) (1/t <sub>SU</sub> —No Feedback)			26.3			21.7		MHz
f <sub>CNT</sub>	Max. Count Frequency (1/t <sub>CNT</sub> —With Feedback)			20			16.7		MHz
t <sub>SU1</sub>	Input Setup Time to CLK	36			43			+ 25	ns
t <sub>SU2</sub>	I/O Setup Time to CLK	38			46			+ 25	ns
t <sub>H</sub>	I or I/O Hold after CLK High	0			0				ns
t <sub>CO</sub>	CLK High to Output Valid			23			25		ns
t <sub>CNT</sub>	Register Output Feedback to Register Input—Internal Path	50			60			+ 25	ns
t <sub>CH</sub>	CLK High Time	17.5			23				ns
t <sub>CL</sub>	CLK Low Time	17.5			23				ns

**ASYNCHRONOUS CLOCK MODE A.C. CHARACTERISTICS** $T_A = 0^\circ\text{C to } 70^\circ\text{C}$ ,  $V_{CC} = 5.0\text{V} \pm 5\%$ , Turbo Bit On<sup>(8)</sup>

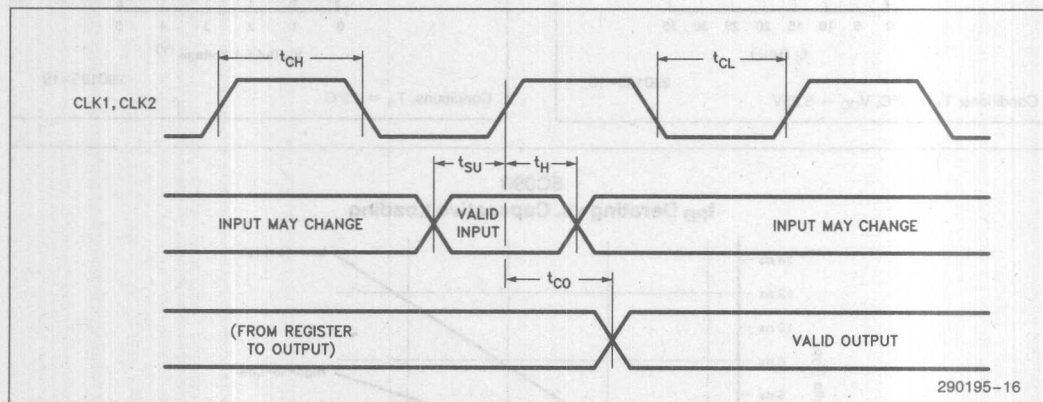
Symbol	Parameter	Device						Non-(11) Turbo Mode	Unit
		5C090-50 EP900-2			5C090-60 EP900				
		Min	Typ	Max	Min	Typ	Max		
f <sub>ACNT</sub>	Max. Count Frequency * (1/t <sub>ACNT</sub> —With Feedback)			20			16.7		MHz
t <sub>ASU1</sub>	Input Setup Time to Asynch. Clock	10			10			+ 25	ns
t <sub>ASU2</sub>	I/O Setup Time to Asynch. Clock	13			15			+ 25	ns
t <sub>AH</sub>	Input or I/O Hold After Asynch. Clock	15			15				ns
t <sub>ACO</sub>	Asynch. CLK to Output Valid			48			59	+ 25	ns
t <sub>ACNT</sub>	Register Output Feedback to Register Input—Internal Path	50			60			+ 25	ns
t <sub>ACH</sub>	Asynch. CLK High Time	17.5			23			+ 25	ns
t <sub>ACL</sub>	Asynch. CLK Low Time	17.5			23			+ 25	ns

## SWITCHING WAVEFORMS

## COMBINATORIAL MODE

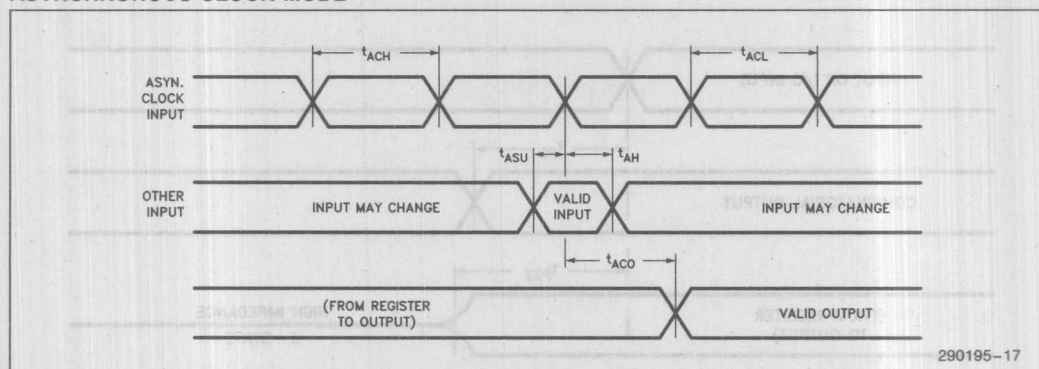


## SYNCHRONOUS CLOCK MODE



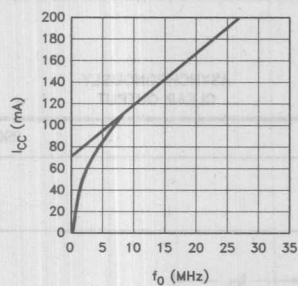
# SWITCHING WAVEFORMS (Continued)

## ASYNCHRONOUS CLOCK MODE



290195-17

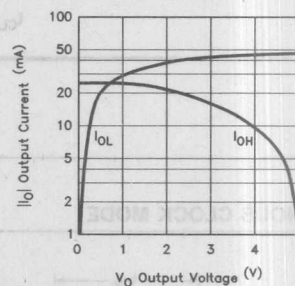
**5C090**  
Current in Relation to Frequency



Conditions:  $T_A = 0^\circ\text{C}$ ,  $V_{CC} = 5.25\text{V}$

290195-18

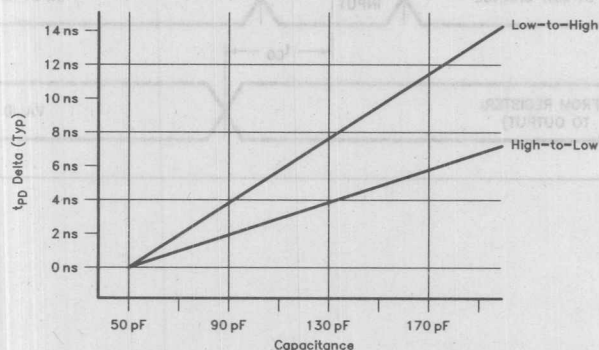
**5C090**  
Output Drive Current in Relation to Voltage



Conditions:  $T_A = 25^\circ\text{C}$

290195-19

**5C090**  
 $t_{PD}$  Derating vs. Capacitive Loading



CONDITIONS:  
 $T_A = 25^\circ\text{C}$ ,  $V_{CC} = 5.0\text{V}$

290195-20

- **Logic Alternative for TTL and 74HC SSI and MSI Logic**
- **48 Macrocells with Programmable I/O Architecture; up to 64 Inputs (16 Dedicated, 48 I/O) or 48 Outputs**
- **High Speed  $t_{PD}$  (max) 70 ns, 20.8 MHz Pipelined, 16.1 MHz w/Feedback**
- **Dual Feedback Signals Allowing I/O Pins to Be Used for Buried Logic and Dedicated Input**
- **Programmable Clock System with Four Synchronous Clocks as well as Asynchronous Clocking Option on All Registers**

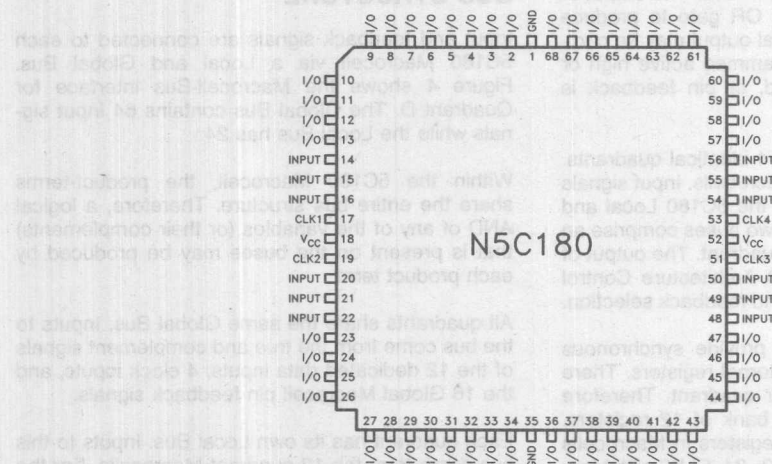
## Configured as D, T, SR or JK Types with Individual Reset Controls

- **Low Power; 100  $\mu$ W Typical Standby Dissipation**
- **Programmable "Security Bit" Allows Total Protection of Proprietary Designs**
- **100% Generically Tested Logic Array**
- **68-Pin J-Lead Chip Carrier**  
(See Packaging Spec., Order #240800-001, Package Type N)
- **100% Compatible with EP1800**

2

The Intel 5C180 EPLD (Erasable Programmable Logic Device) is a CHMOS, 48-macrocell, general-purpose PLD. This user-customizable Logic Device is available in a 68-pin PLCC package and has the benefits of low power and increased flexibility.

The 5C180 EPLD uses CHMOS EPROM (floating gate) cells as logic control elements instead of fuses. Use of Intel's advanced CHMOS II-E EPROM process technology enables greater logic densities to be achieved with superior speed and power performance. The EPROM technology enables these devices to be 100% factory tested by the programming and the erasure of all the EPROM logic control elements in the device.



290111-1

Figure 1. Pin Configurations



The architecture of the 5C180 is based on the "Sum of Products" PLA (Programmable Logic Array) structure with a programmable AND array feeding into a fixed OR array. The 48 macrocells of the 5C180 can be partitioned into 4 identical quadrants each containing 12 macrocells. This device makes use of a segmented PLA structure with local and global bus structures to provide for increased performance and greater device utilization. The 5C180 has unique architectural features that allow programming of all 48 registers to D, T, SR or JK configurations without sacrificing product terms. These registers can be either clocked asynchronously or in banks with four synchronous clocks. In addition, the 16 global macrocells have two independent feedback paths to the array that allow for buried logic implementation together with use of the I/O pin for input functions.

## ARCHITECTURE DESCRIPTION

Externally, the 5C180 provides 12 dedicated data inputs, 4 synchronous clock inputs, and 48 I/O pins which may be individually programmed for input, output, or bi-directional operation.

The Block Diagram is shown in Figure 2 with pin numbers for the PLCC package. The internal architecture is organized in familiar sum-of-products (AND-OR) structure. The 5C180 houses a total of 480 product terms distributed among 48 Macrocells. The basic Macrocell structure is shown in Figure 3. Input and feedback signals are selectively connected to product terms via EPROM cells. The output of the AND array feeds a fixed OR gate to produce sum-of-products logic. The final output may be combinatorial or registered, programmed active high or low. Combinatorial, registered, or pin feedback is also user-defined.

The 5C180 is partitioned into 4 identical quadrants. Each quadrant contains 12 Macrocells. Input signals to the Macrocells come from the 5C180 Local and Global bus structures. These two buses comprise an 88-input AND array for each quadrant. The output of each Macrocell feeds an I/O Architecture Control Block which contains output and feedback selection.

Four dedicated clock inputs provide synchronous clock signals to the 5C180 internal registers. There is one synchronous clock per quadrant. Therefore each clock signal controls a bank of 12 registers. CLK1 may be connected to registers in Macrocells 1-12, CLK2 with Macrocells 13-24, CLK3 with Macrocells 25-36, and CLK4 with Macrocells 37-48. With synchronous clocks, the flip-flops are positive edge triggered. Both true and complement signals for each dedicated clock input may also be used

within the AND array. All 48 internal registers may be individually programmed for synchronous or asynchronous clocking. Asynchronous clocking is possible via a Macrocell product term. Clock inputs not used for synchronous clock signals may be used as global bus inputs.

## Invert Select EPROM Bit

The Invert Select EPROM bit is used to invert the product term input into the register. This applies to all inputs including double inputs on JK and SR registers. The invert option allows the highest possible logic utilization by use of deMorgan logic inversion.

At each intersecting point in the logic array there exists an EPROM-type programmable connection. Initially, all connections are complete. This means that both the true and complement of all inputs are connected to each product term. Connections are opened during the programming process. Therefore any product term can be connected to the true or complement of any input. When both the true and complement connections of any input are left intact, a logical false results on the output of the AND gate. If both the true and complement connections of any input are programmed open, then a logical "don't care" results for that input. If all inputs for a product term are programmed open, then a logical true results on the output of the AND gate.

## BUS STRUCTURE

Input and feedback signals are connected to each 5C180 Macrocell via a Local and Global Bus. Figure 4 shows the Macrocell-Bus interface for Quadrant D. The Global Bus contains 64 input signals while the Local Bus has 24.

Within the 5C180 Macrocell, the product-terms share the entire bus structure. Therefore, a logical AND of any of the variables (or their complements) that is present on the buses may be produced by each product term.

All quadrants share the same Global Bus. Inputs to the bus come from the true and complement signals of the 12 dedicated data inputs, 4 clock inputs, and the 16 Global Macrocell pin feedback signals.

Each quadrant has its own Local Bus. Inputs to this bus come from the 12 quadrant Macrocells. For the eight Local Macrocells, the signals can be either from the Macrocell internal logic or from the pin. For the four Global Macrocells, the signals come from the Macrocell internal logic only.

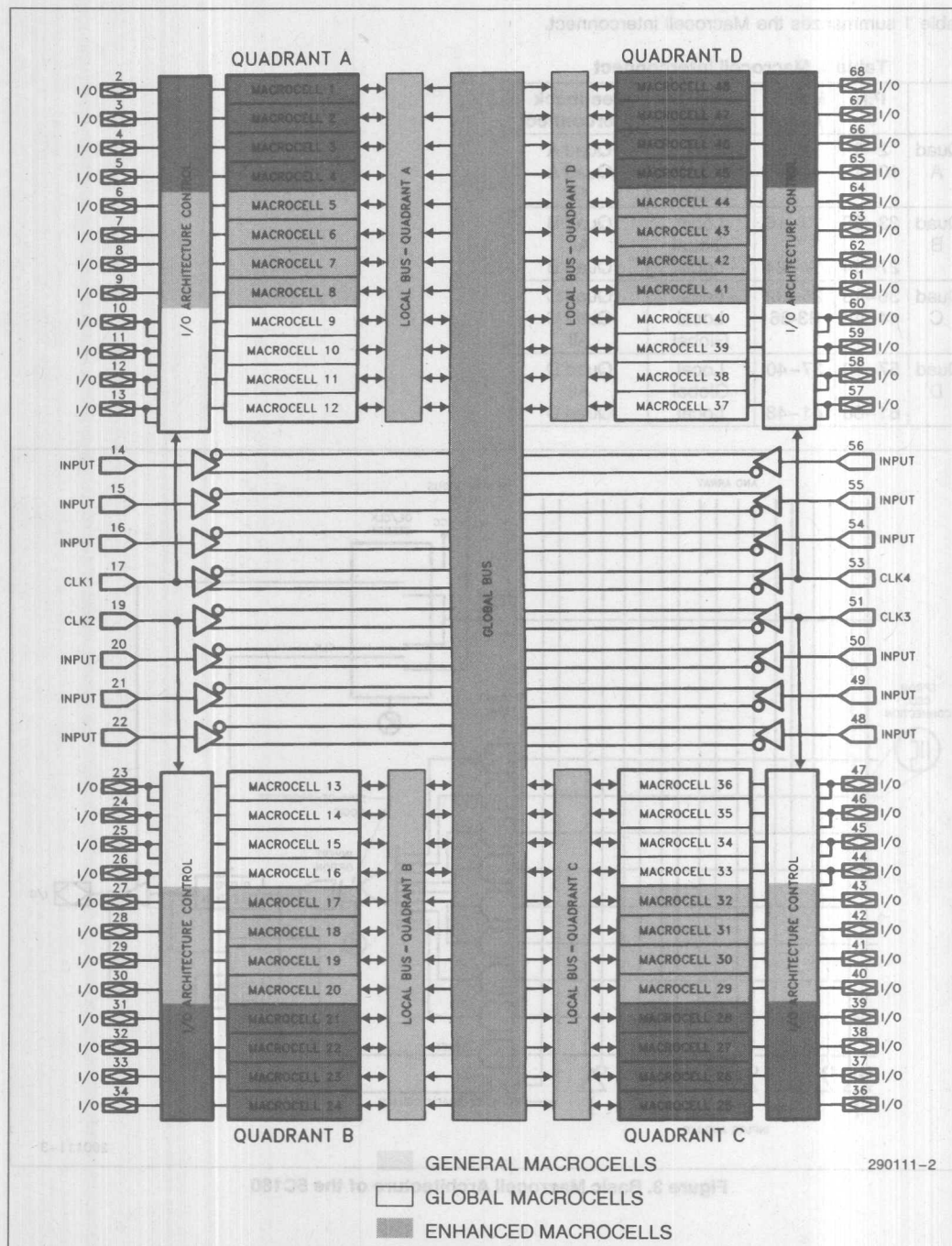
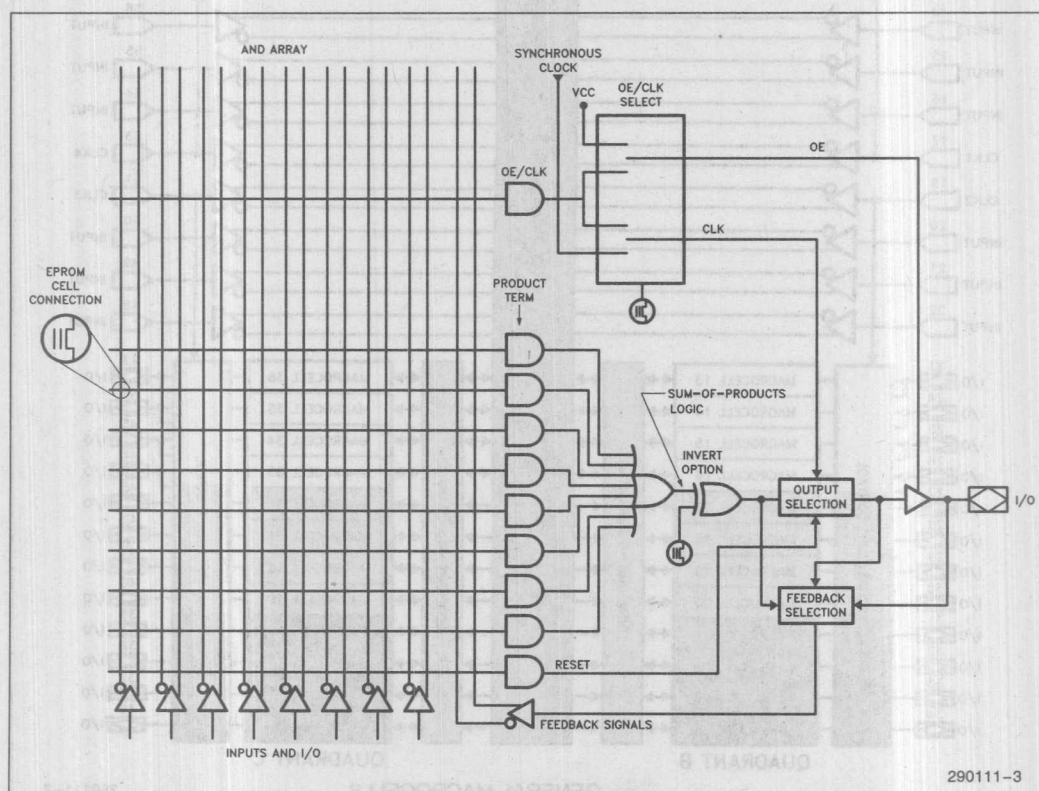


Figure 2. 5C180 Block Diagram—PLCC Package

Table 1 summarizes the Macrocell interconnect.

**Table 1. Macrocell Interconnect**

	Pin #	Macro-cell #	Feedback Structure	Feedback Interconnect
Quad A	2-9	1-8	Local	Quad A
	10-13	9-12	Local Global	Quad A All
Quad B	23-26	13-16	Local	Quad B
	27-34	17-24	Global Local	All Quad B
Quad C	36-43	25-32	Local	Quad C
	44-47	33-36	Local Global	Quad C All
Quad D	57-60	37-40	Local	Quad D
	61-68	41-48	Global Local	All Quad D



**Figure 3. Basic Macrocell Architecture of the 5C180**

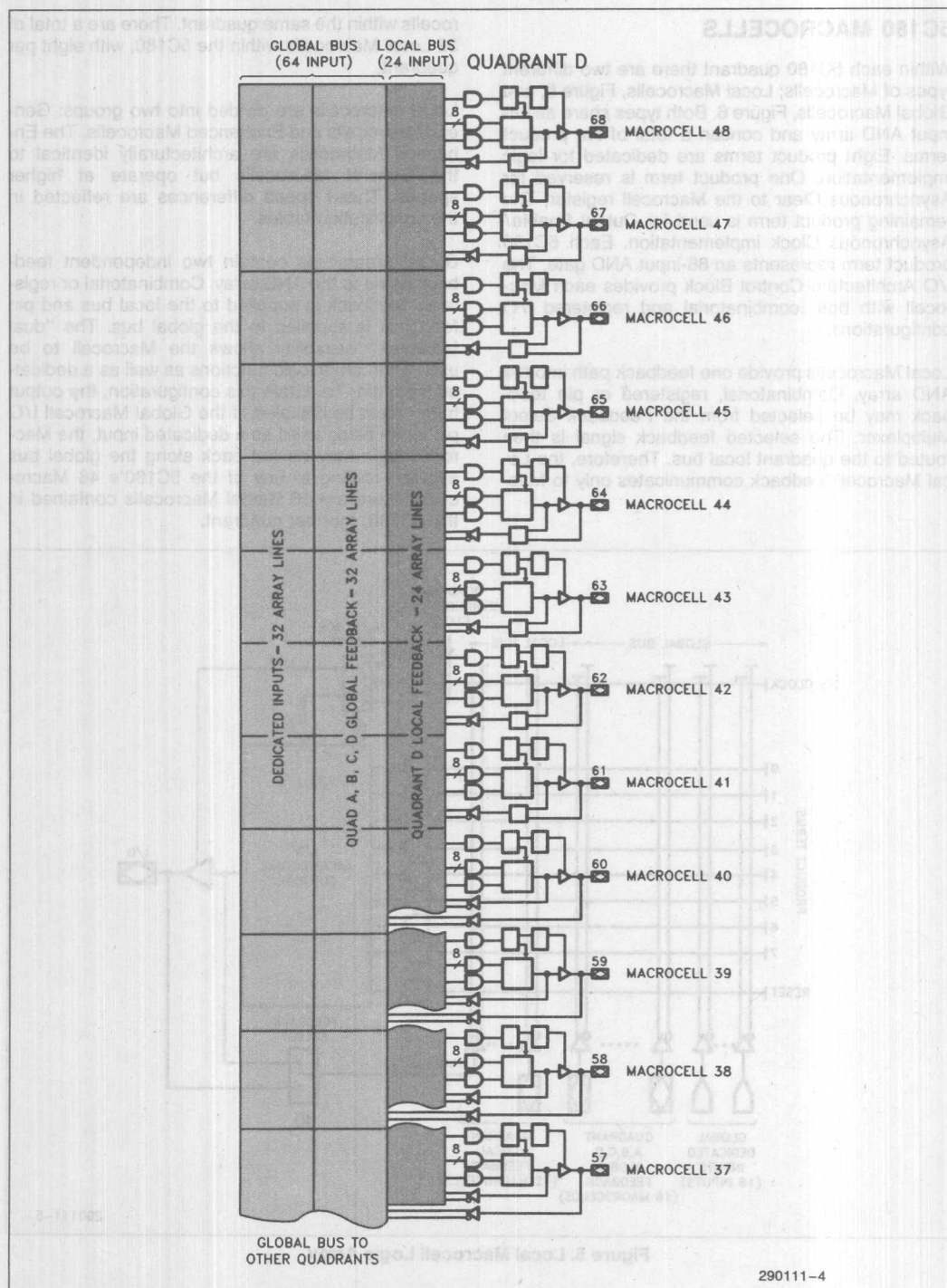


Figure 4. Quadrant "D" Bus Interface



## 5C180 MACROCELLS

Within each 5C180 quadrant there are two different types of Macrocells; Local Macrocells, Figure 5, and Global Macrocells, Figure 6. Both types share an 88-input AND array and contain a total of ten product terms. Eight product terms are dedicated for logic implementation. One product term is reserved for Asynchronous Clear to the Macrocell register. The remaining product term is used for Output Enable/Asynchronous Clock implementation. Each 5C180 product term represents an 88-input AND gate. The I/O Architecture Control Block provides each Macrocell with both combinatorial and registered I/O configurations.

Local Macrocells provide one feedback path into the AND array. Combinatorial, registered or pin feedback may be selected from the Feedback Select Multiplexer. The selected feedback signal is then routed to the quadrant local bus. Therefore, the Local Macrocell feedback communicates only to Macro-

cells within the same quadrant. There are a total of 32 Local Macrocells within the 5C180, with eight per quadrant.

Local macrocells are divided into two groups: General Macrocells and Enhanced Macrocells. The Enhanced Macrocells are architecturally identical to the General Macrocells but operate at higher speeds. These speed differences are reflected in the specification tables.

Global Macrocells contain two independent feedback paths to the AND array. Combinatorial or registered feedback is supplied to the local bus and pin feedback is supplied to the global bus. The "dual feedback" capability allows the Macrocell to be used for internal logic functions as well as a dedicated input pin. To obtain this configuration, the output buffer must be disabled. If the Global Macrocell I/O pin is not being used as a dedicated input, the Macrocell logic may be fed back along the global bus allowing routing to any of the 5C180's 48 Macrocells. There are 16 Global Macrocells contained in the 5C180, four per quadrant.

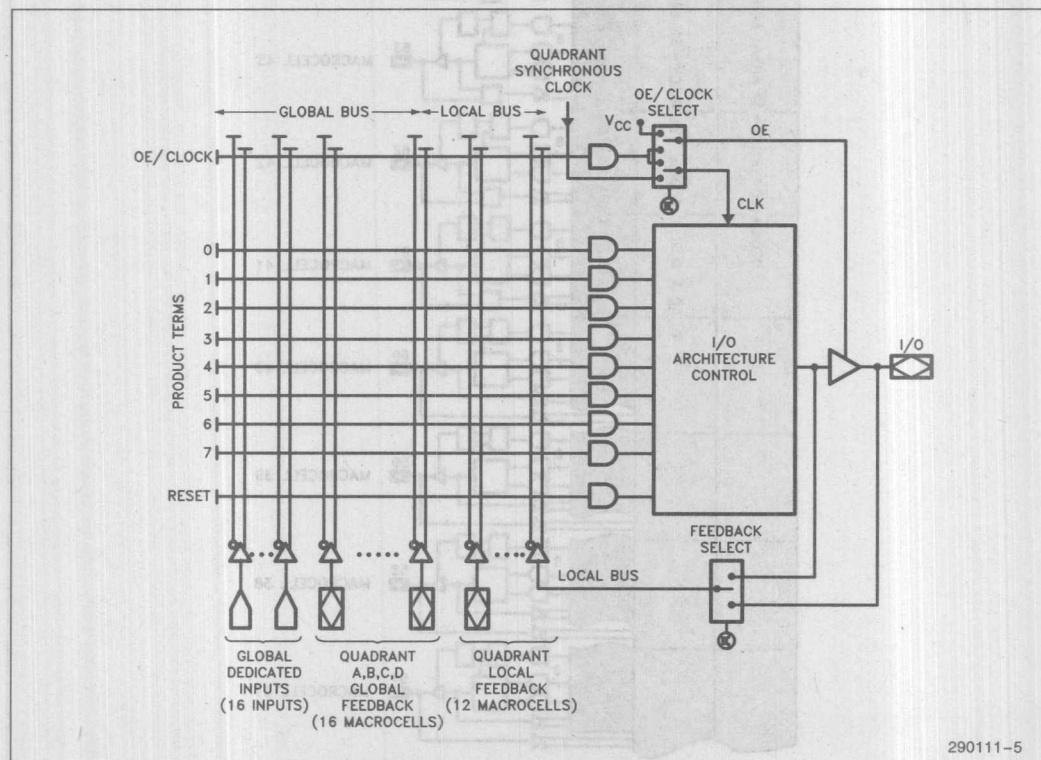


Figure 5. Local Macrocell Logic Array

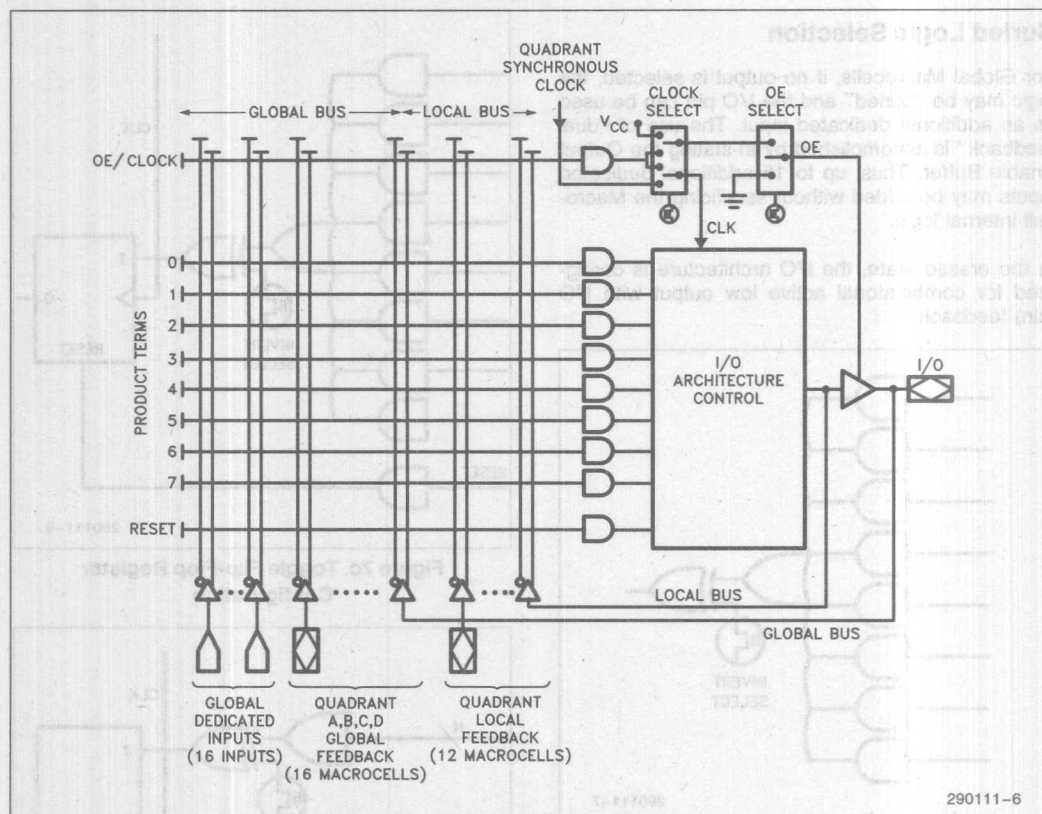


Figure 6. Global Macrocell Logic Array

## MACROCELL LOGIC CONFIGURATIONS

### Combinatorial Selection

In the Combinatorial configuration, eight product terms are ORed together to generate the output signal. The Invert Select EPROM bit controls output polarity and the Output Enable buffer is product-term controlled. The Feedback Select allows the user to choose combinatorial, I/O (pin) or no feedback to the respective local and global buses.

### REGISTER SELECTION

The advanced I/O architecture of the 5C180 allows four different register types along with combinatorial output as illustrated in Figures 8a–8e. The register types include a T, D, JK, or SR Flip-Flop and each Macrocell I/O structure may be independently configured. In addition, all registers have an individual asynchronous RESET control from a dedicated

product term derived in the AND array. When this dedicated product term is a logical one, the Macrocell register is immediately cleared to a logical zero independent of the register clock. The RESET function occurs automatically on power-up.

The four different register types shown in Figures 7b–7e are described below:

#### D- or T-type Flip-Flops

When either a D- or T-type Flip-Flop is configured as part of the I/O structure, all eight of the product terms into the Macrocell are ORed together and fed into the register input.

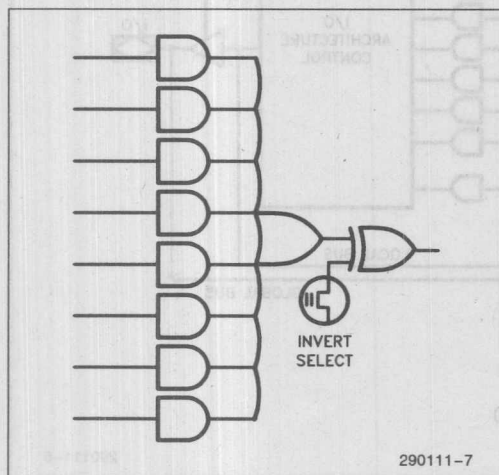
#### JK or SR Registers

When either a JK or SR register is configured, the eight product terms are shared among two OR gates (one for the J or S input and the other for the K or R input). The allocation for these product terms for each of the register inputs is optimized by the iPLDS II development software.

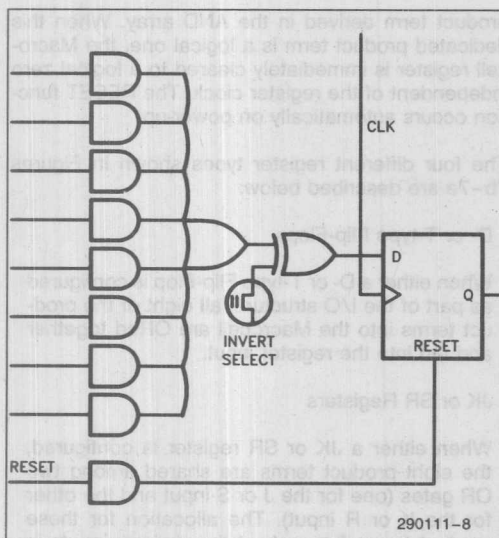
## Buried Logic Selection

For Global Macrocells, if no output is selected, the logic may be "buried" and the I/O pin can be used as an additional dedicated input. The use of "dual feedback" is accomplished by tri-stating the Output Enable Buffer. Thus, up to 16 additional dedicated inputs may be added without sacrificing the Macrocell internal logic.

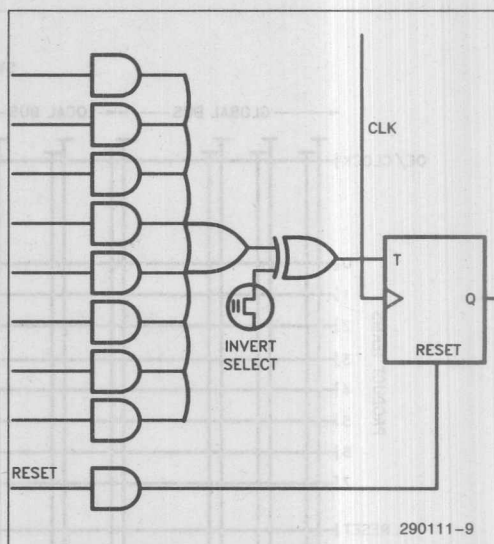
In the erased state, the I/O architecture is configured for combinatorial active low output with I/O (pin) feedback.



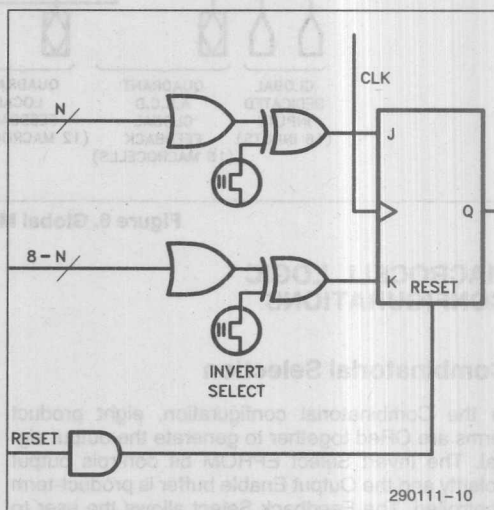
### Figure 7a. Combinatorial I/O Configuration



### Figure 7b. D-Type Flip-Flop Register Configuration



### Figure 7c. Toggle Flip-Flop Register Configuration



### Figure 7d. JK Flip-Flop Register Configuration

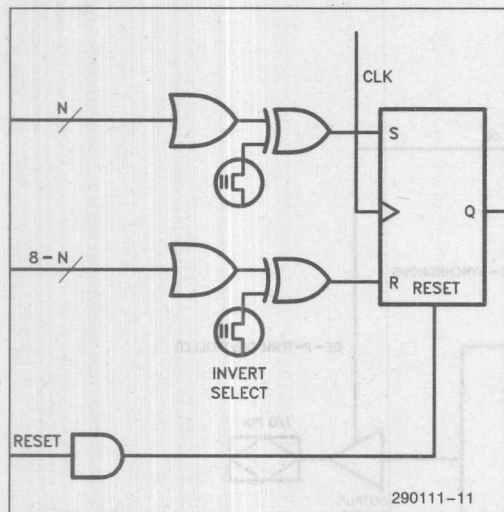


Figure 7e. SR Flip-Flop Register Configuration

### MACROCELL OE/CLK SELECT

Each 5C180 register may be clocked synchronously or asynchronously. Figure 8a and 8b shows the modes of operation provided by the OE/CLK Select Multiplexers for both Local and Global Macrocells.

The operation of each multiplexer is controlled by EPROM bits and may be individually configured for each 5C180 Macrocell.

In Mode 0, the three-state output buffer is controlled by a single product term. If the output of the AND gate is a logical true then the output buffer is enabled. If a logical false resides on the output of the AND gate then the output buffer is seen as high impedance. In this mode the Macrocell flip-flop may be clocked by its quadrant synchronous clock input. In the erased state, the 5C180 is configured as Mode 0.

In Mode 1, the Output Buffer is always enabled. The Macrocell flip-flop now may be triggered from an asynchronous clock signal generated by the Macrocell product term. This mode allows individual clocking of flip-flops from any available signal in the quadrant AND array. Because both true and complement signals reside in the AND array, the flip-flops may be clocked by positive- or negative-going signals at any input pin. With the clock now controlled by a product term, gate clock structures are also possible.

In Modes 2 and 3, the Output Buffer is always disabled. The Macrocell flip-flop may still be triggered from clock signals generated from the Macrocell product term or asynchronous clocks. This mode is only possible for Global Macrocells.



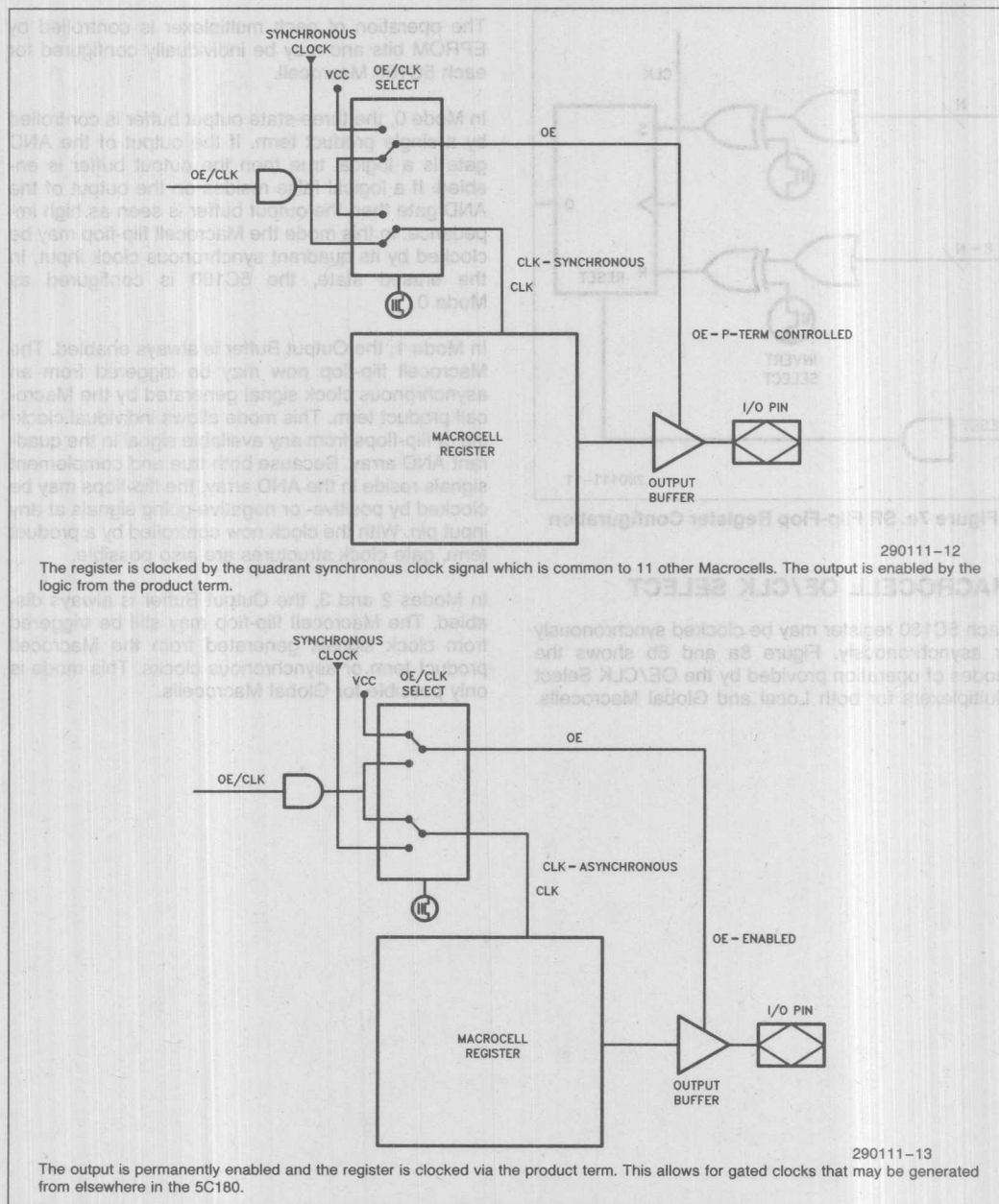
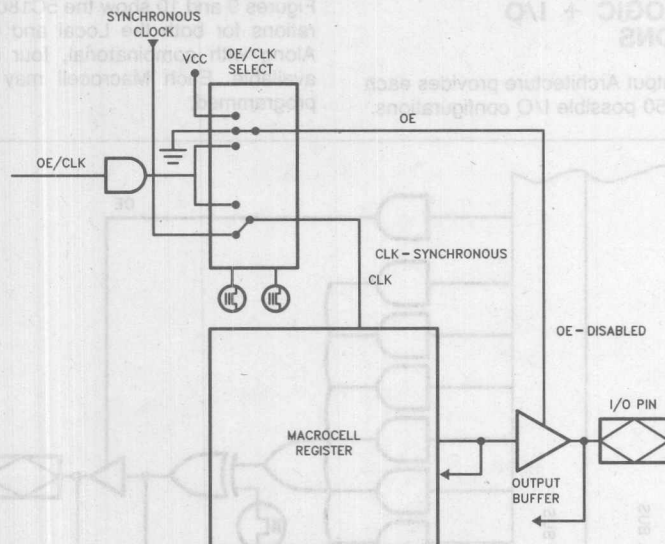
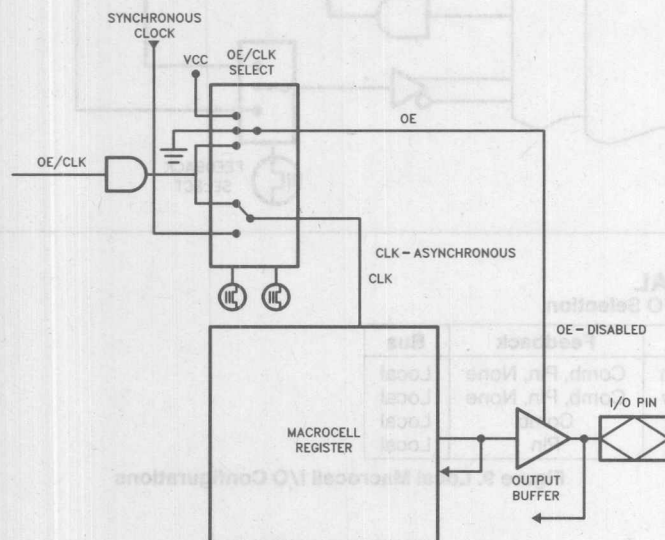


Figure 8a. Local Macrocell OE/CLK Selection



290111-14

The output is permanently disabled and the register clocked by the quadrant synchronous clock signal. The pin can be used as an input while the register or combinational output can be fed back.



290111-15

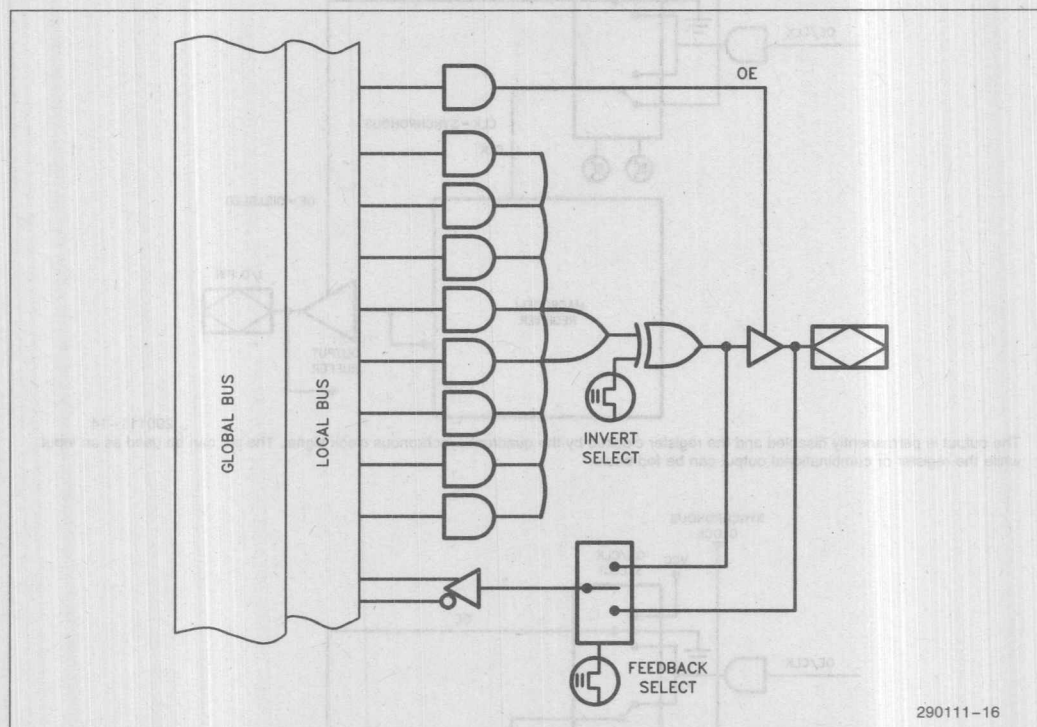
The output is permanently disabled and the register is clocked via the product term. This allows gated clocks that may be generated elsewhere in the 5C180. The pin can be used as an input while the register or combinational output can be fed back.

**Figure 8b. Global Macrocell Additional OE/CLK Selection**

## MACROCELL LOGIC + I/O CONFIGURATIONS

The 5C180 Input/Output Architecture provides each Macrocell with over 50 possible I/O configurations.

Figures 9 and 10 show the 5C180 basic I/O configurations for both the Local and Global Macrocells. Along with combinatorial, four register types are available. Each Macrocell may be independently programmed.

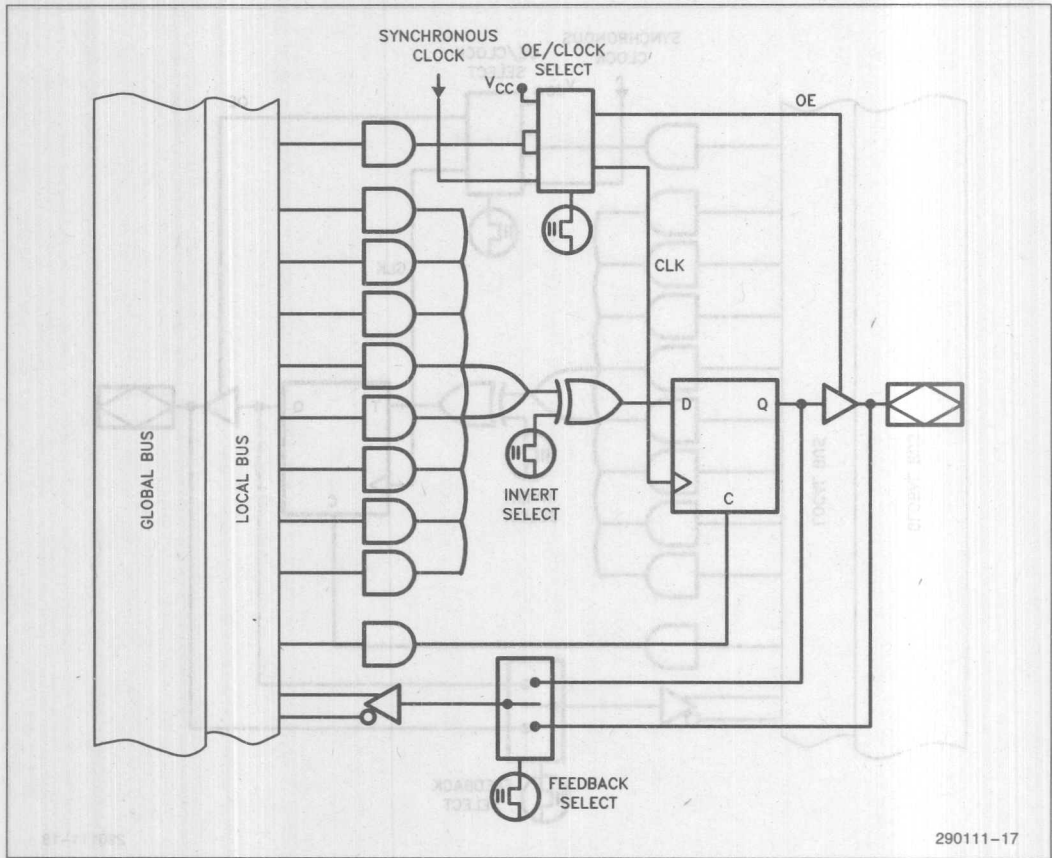


## COMBINATORIAL

### I/O Selection

Output/Polarity	Feedback	Bus
Combinatorial/High	Comb, Pin, None	Local
Combinatorial/Low	Comb, Pin, None	Local
None	Comb	Local
None	Pin	Local

Figure 9. Local Macrocell I/O Configurations



2

**D-TYPE FLIP-FLOP**  
I/O Selection

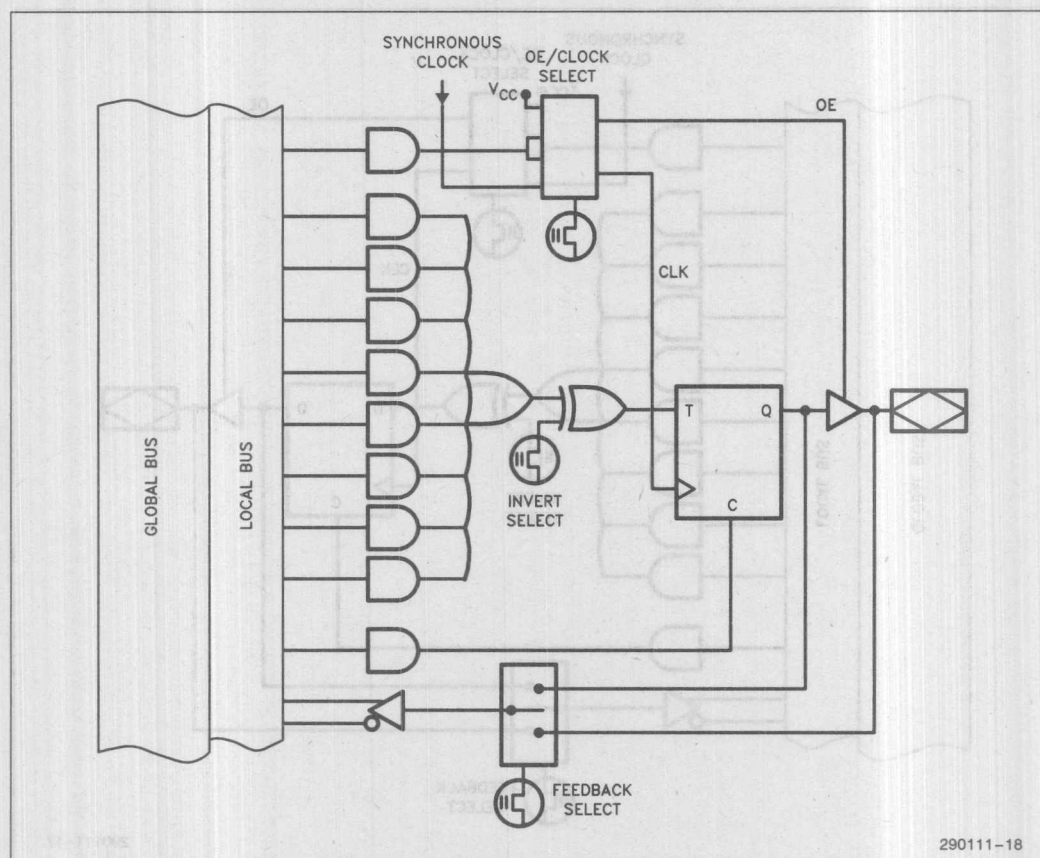
Output/Polarity	Feedback	Bus
D-Register/High	D-Register, Pin, None	Local
D-Register/Low	D-Register, Pin, None	Local
None	D-Register	Local
None	Pin	Local

**Function Table**

D	$Q_n$	$Q_{n+1}$
0	0	0
0	1	0
1	0	1
1	1	1

Figure 9. Local Macrocell I/O Configurations (Continued)





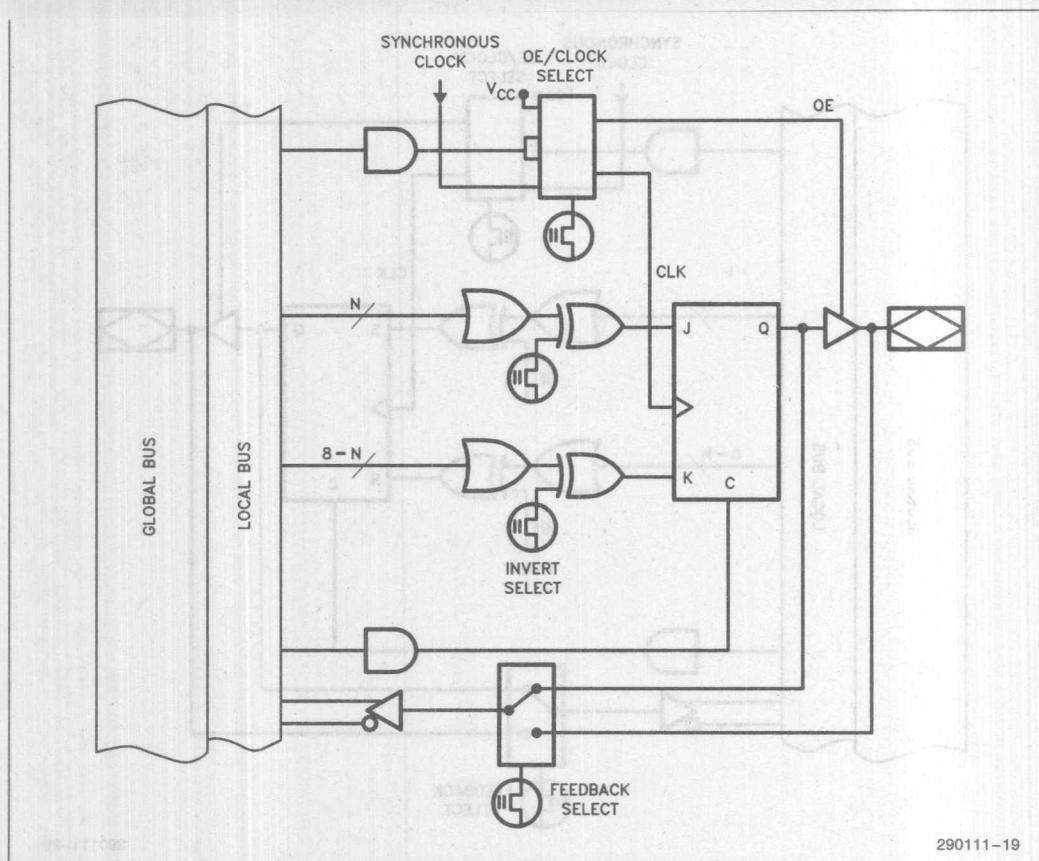
### TOGGLE FLIP-FLOP I/O Selection

Output/Polarity	Feedback	Bus
T-Register/High	T-Register, Pin, None	Local
T-Register/Low	T-Register, Pin, None	Local
None	T-Register	Local
None	Pin	Local

Function Table

T	$Q_n$	$Q_{n+1}$
0	0	0
0	1	1
1	0	1
1	1	0

Figure 9. Local Macrocell I/O Configurations (Continued)



## JK FLIP-FLOP

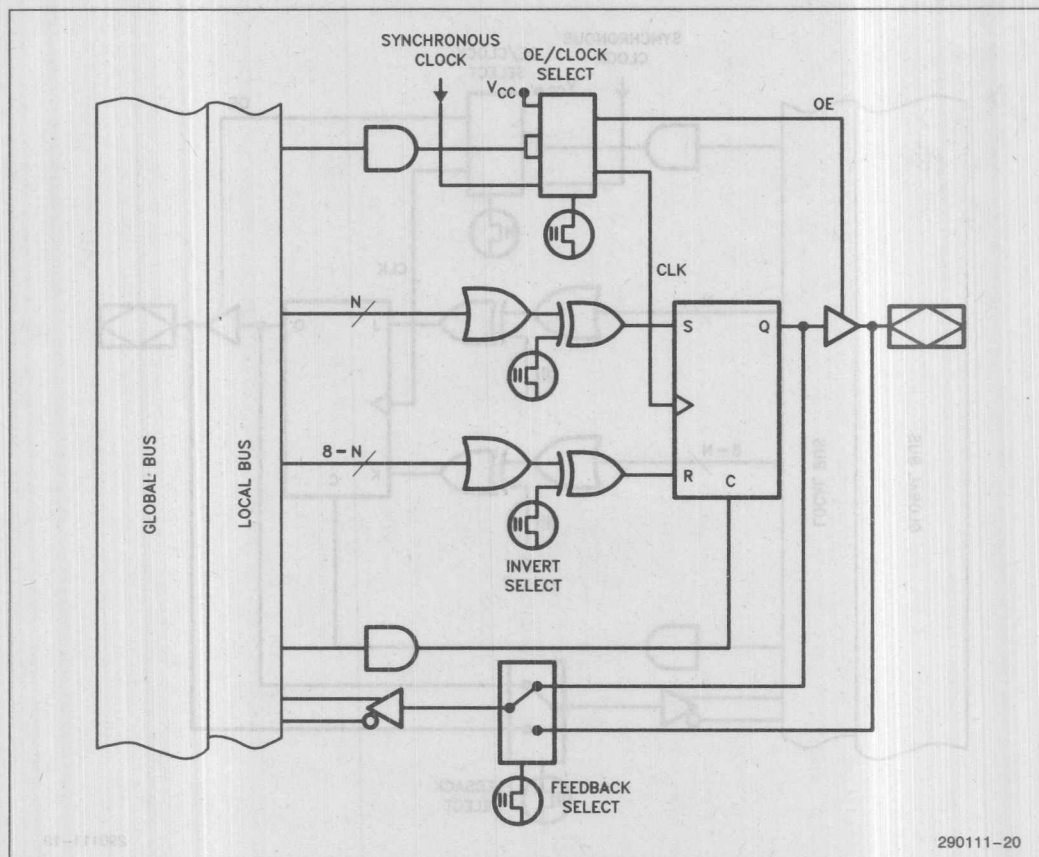
## I/O Selection

Output/Polarity	Feedback	Bus
JK Register/High	JK Register, None	Local
JK Register/Low	JK Register, None	Local
None	JK Register	Local

## Function Table

J	K	$Q_n$	$Q_{n+1}$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

**Figure 9. Local Macrocell I/O Configurations (Continued)**



# SR FLIP-FLOP

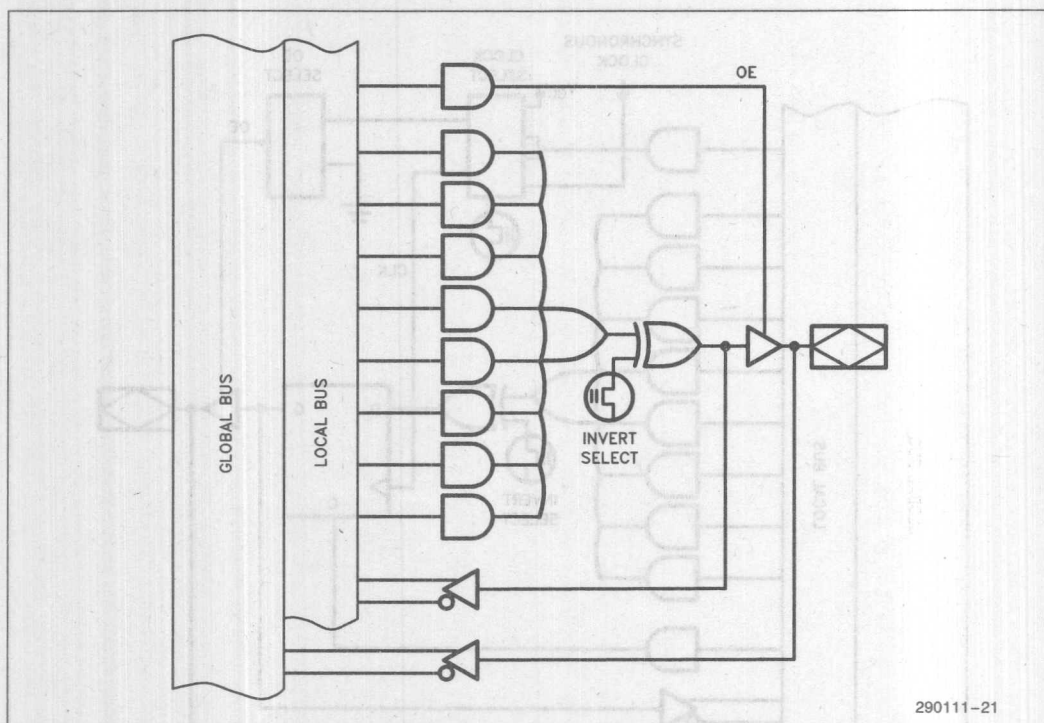
## I/O Selection

Output/Polarity	Feedback	Bus
SR Register/High	SR Register, None	Local
SR Register/Low	SR Register, None	Local
None	SR Register	Local

## Function Table

S	R	$Q_n$	$Q_{n+1}$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1

Figure 9. Local Macrocell I/O Configurations (Continued)



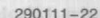
2

# **COMBINATORIAL** I/O Selection

Output/Polarity	Feedback	Bus
Combinatorial/High	Comb, Pin, None	Local, Global
Combinatorial/Low	Comb, Pin, None	Local, Global
None	Comb	Local, Global
None	Pin	Global
None	Comb/Pin	Local/Global

Figure 10. Global Macrocell I/O Configurations





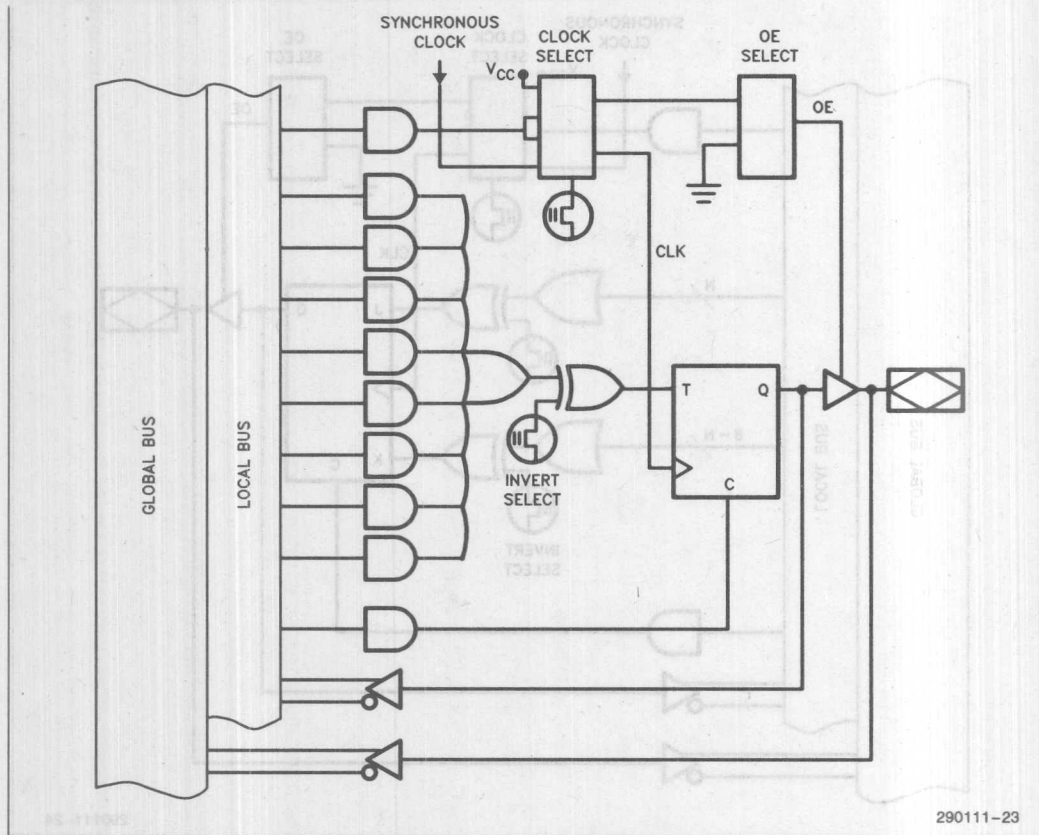
290111-22	Feedback	Bus
-----------	----------	-----

Global Figure 10 shows

Global | Global St. Louis

Global Figure 10 shows

Global | Global St. Louis



2

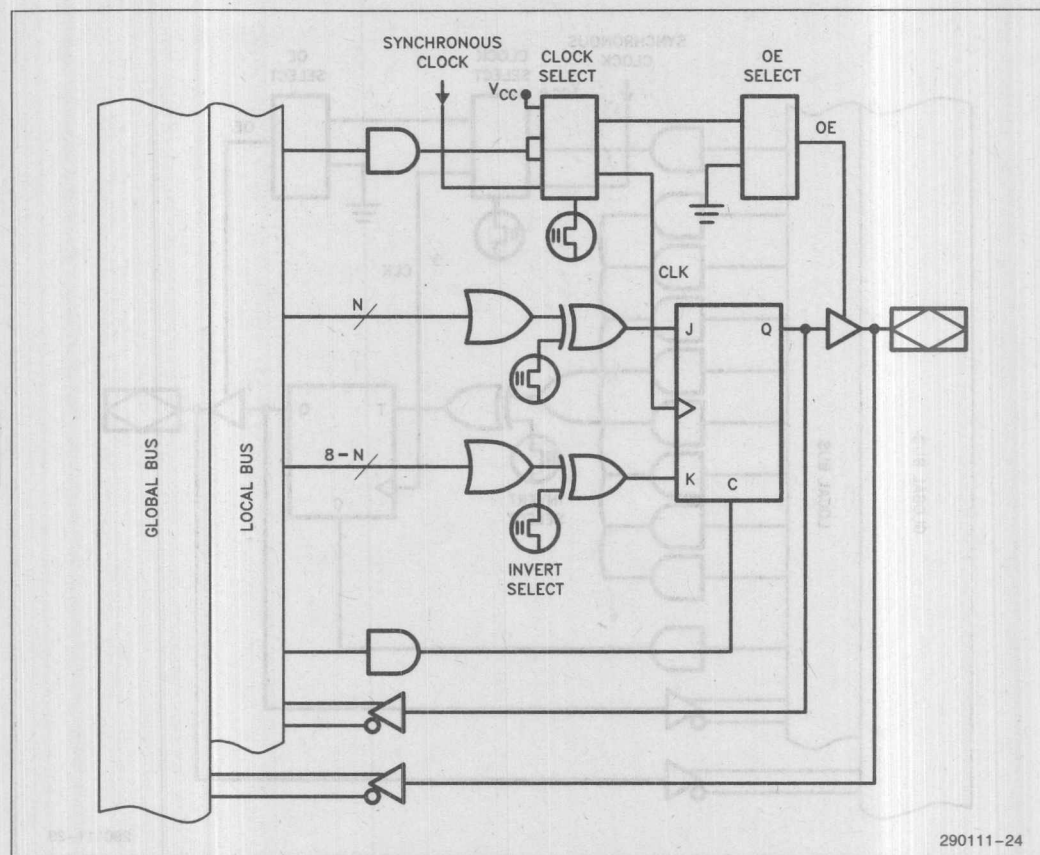
### TOGGLE FLIP-FLOP I/O Selection

Output/Polarity	Feedback	Bus
T-Register/High	T-Register, Pin, None	Local, Global
T-Register/Low	T-Register, Pin, None	Local, Global
None	T-Register	Local, Global
None	Pin	Global
None	T-Register/Pin	Local/Global

#### Function Table

T	$Q_n$	$Q_{n+1}$
0	0	0
0	1	1
1	0	1
1	1	0

Figure 10. Global Macrocell I/O Configurations (Continued)



## JK FLIP-FLOP

### I/O Selection

Output/Polarity	Feedback	Bus
JK Register/High	JK Register, None	Local, Global
JK Register/Low	JK Register, None	Local, Global
None	JK Register	Local
None	JK Register/Pin	Local/Global

### Function Table

J	K	Q <sub>n</sub>	Q <sub>n+1</sub>
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Figure 10. Global Macrocell I/O Configurations (Continued)





## AUTOMATIC STAND-BY MODE

The 5C180 contains a programmable bit, the Turbo Bit, that optimizes operation for speed or for power savings. When the Turbo Bit is programmed (TURBO = ON), the device is optimized for maximum speed. When the Turbo Bit is not programmed (TURBO = OFF), the device is optimized for power savings by entering standby mode during periods of inactivity.

Figure 11 shows the device entering standby mode approximately 100 ns after the last input transition. When the next input transition is detected, the device returns to active mode. Wakeup time adds an additional 30 ns to the propagation delay through the device as measured from the first input. No delay will occur if an output is dependent on more than one input and the last of the inputs changes after the device has returned to active mode.

After erasure, the Turbo Bit is unprogrammed (OFF); automatic standby mode is enabled. When the Turbo Bit is programmed (ON), the device never enters standby mode.

## Erased-State Configuration

Prior to programming, the I/O structure is configured for combinatorial active low output with input (pin) feedback.

## PROGRAMMING CHARACTERISTICS

Initially, all the EPROM control bits of the 5C180 are connected. Each of the connected control bits are selectively disconnected by programming the EPROM cells into their "on" state. Programming voltage and waveform specifications are available by request from Intel to support programming of the 5C180.

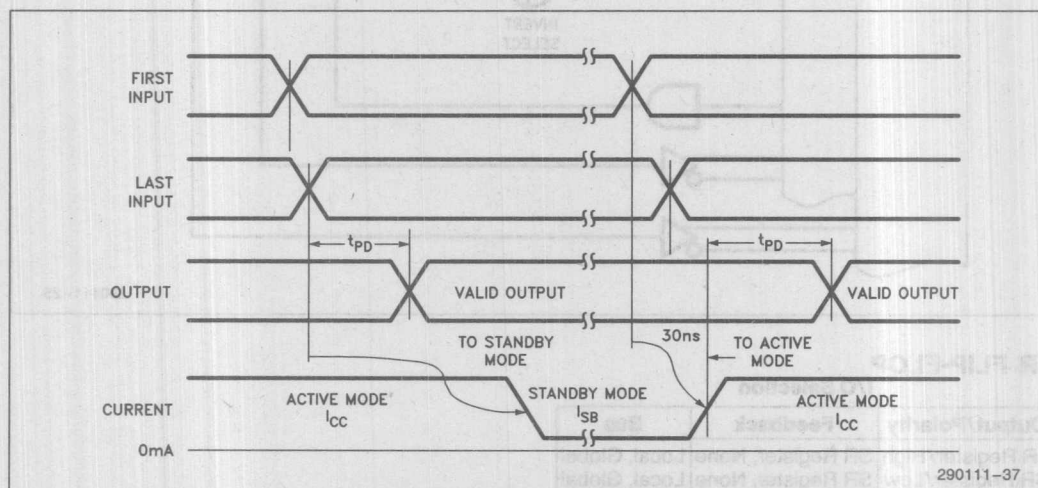


Figure 11. 5C180 Standby and Active Mode Transitions

1	2	3	4
0	0	0	0
1	1	0	0
0	0	1	0
0	1	1	0
1	0	0	1
1	1	0	1

## Intelligent Programming Algorithm

The 5C180 supports the Intelligent Programming Algorithm which rapidly programs Intel PLDs using an efficient and reliable method. The Intelligent Programming Algorithm is particularly suited to the production programming environment. This method ensures reliability as the incremental program margin of each bit is continually monitored to determine when the bit has been successfully programmed.

## FUNCTIONAL TESTING

Since the logical operation of the 5C180 is controlled by EPROM elements, the device is completely testable. Each programmable EPROM bit controlling the internal logic is tested using application-independent test program patterns. After testing, the devices are erased before shipment to customers. No post-programming tests of the EPROM array are required.

The testability and reliability of EPROM-based programmable logic devices is an important feature over similar devices based on fuse technology. Fuse-based programmable logic devices require a use to perform post-programming tests to insure proper programming. These tests must be done at the device level because of the cumulative error effect. For example, a board containing ten devices each possessing a 2% device fallout translates into an 18% fallout at the board level (it should be noted that programming fallout of fuse-based programmable logic devices is typically 2% or higher).

## DESIGN RECOMMENDATIONS

For proper operation, it is recommended that all input and output pins be constrained to the voltage range  $GND < (V_{IN} \text{ or } V_{OUT}) < V_{CC}$ . Unused inputs and I/Os should be tied to  $V_{CC}$  or GND to minimize device power consumption. Reserved pins (as indicated in the logic compiler REPORT file) should be left floating (no connect) so that the pin can attain the appropriate logic level. A power supply decoupling capacitor of at least  $0.2 \mu f$  must be connected directly between  $V_{CC}$  and GND.

As with all CMOS devices, ESD handling procedures should be used with this device to prevent damage during programming, assembly, and test.

## DESIGN SECURITY

A single EPROM bit provides a programmable design security feature that controls the access to the data programmed into the device. If this bit is set, a proprietary design within the device cannot be copied. This EPROM security bit enables a higher degree of design security than fused-based devices since programmed data within EPROM cells is invisible even to microscopic evaluation. The EPROM security bit, along with all the other EPROM control bits, will be reset by erasing the device.

## LATCH-UP IMMUNITY

All of the input, I/O, and clock pins of the 5C180 have been designed to resist latch-up which is inherent in inferior CMOS structures. The 5C180 is designed with Intel's proprietary CHMOS II-E EPROM process. Thus, each of the 5C180 pins will not experience latch-up with currents up to  $\pm 100 \text{ mA}$  and voltages ranging from  $-1V$  to  $(V_{CC} + 1V)$ . Furthermore, the programming pin is designed to resist latch-up to the 13.5V maximum device limit.

## SOFTWARE SUPPORT

Full logic compilation and functional simulation for the 5C180 is supported by PLDshell Plus™ software. The GUPI Logic-18 provides programming support on Intel programmers.

PLDshell Plus design software is Intel's new, user-friendly design tool for  $\mu$ PLD design. PLDshell Plus allows users to incorporate their preferred text editor, programming software, and additional design tools into an easy-to-use, menued design environment that includes Intel's PLDasm™ logic compiler and simulation software along with disassembly, conversion, and translation utilities. The PLDasm compiler and simulator software accepts industry-standard PDS source files that express designs as Boolean equations, truth tables, or state machines. On-line help, datasheet briefs, technical notes, and error message information, along with waveform viewing/printing capability make the design task as easy as possible. PLDshell Plus software is available from Intel Literature channels or from your local Intel sales representative.

Tools that support schematic capture and timing simulation for the 5C180 are available. Support under iPLS II is still available. Please refer to the "Development Tools" section of the Programmable Logic handbook.

I/O, Logic Devices, STAG, etc. Please refer to the "Third-Party Support" lists in the *Programmable Logic* handbook for complete information and vendor contacts.

door contacts.

Code	Package	Operating Range
70	PLCC	Commercial
75	PLCC	Commercial
90	PLCC	Commercial

2-232

**ABSOLUTE MAXIMUM RATINGS\***

Symbol	Parameter	Min	Max	Units
$V_{CC}$	Supply Voltage <sup>(1)</sup>	-2.0	7.0	V
$V_{PP}$	Programming Supply Voltage <sup>(1)</sup>	-2.0	13.5	V
$V_I$	DC Input Voltage <sup>(1)(2)</sup>	-0.5	$V_{CC} + 0.5$	V
$t_{stg}$	Storage Temperature	-65	+150	°C
$t_{amb}$	Ambient Temperature <sup>(3)</sup>	-10	+85	°C

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

\*WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.

**NOTES:**

1. Voltages with respect to ground.
2. Minimum DC input is -0.5V. During transitions, the inputs may undershoot to -2.0V or overshoot to +7.0V for periods less than 20 ns under no load conditions.
3. Under bias. Extended temperature versions are also available.

**RECOMMENDED OPERATING CONDITIONS**

Symbol	Parameter	Min	Max	Units
$V_{CC}$	Supply Voltage	4.75	5.25	V
$V_{IN}$	Input Voltage	0	$V_{CC}$	V
$V_O$	Output Voltage	0	$V_{CC}$	V
$T_A$	Operating Temperature	0	+70	°C
$t_R^{(4)}$	Input Rise Time		500	ns
$t_F^{(4)}$	Input Fall Time		500	ns

**NOTE:**

4.  $t_R$  and  $t_F$  for clocks is 250 ns.

**PACKAGE/TECHNOLOGY SPECIFICATIONS**

Description	Specification
$\theta_{JA}$ —Junction-to-Ambient Thermal Resistance	42°C/W—CerQUAD 38°C/W—PLCC 28.5°C/W—PGA
$\theta_{JC}$ —Junction-to-Case Thermal Resistance	14°C/W—CerQUAD 14°C/W—PLCC 14°C/W—PGA
$I_{CC}$ Hot—Ambient @70°C	140 mA
$I_{CC}$ Typical—Ambient @25°C	140 mA
Process	CHMOS IIE, PX 24

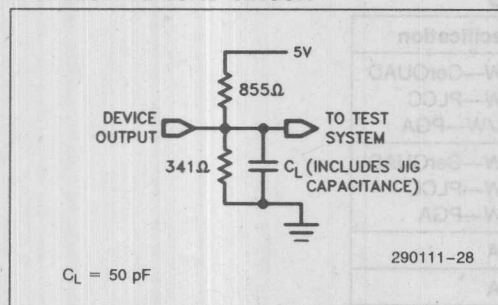
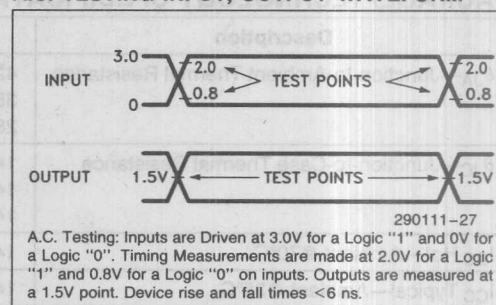


**D.C. CHARACTERISTICS**  $T_A = 0^\circ \text{ to } +70^\circ\text{C}$ ,  $V_{CC} = 5V \pm 5\%$ 

Symbol	Parameter/Test Conditions	Min	Typ	Max	Unit
$V_{IH}^{(5)}$	High Level Input Voltage	2.0		$V_{CC} + 0.3$	V
$V_{IL}^{(5)}$	Low Level Input Voltage	-0.3		0.8	V
$V_{OH}^{(6)}$	High Level Output Voltage $I_O = -4.0 \text{ mA D.C.}, V_{CC} = \text{min.}$	2.4			V
$V_{OL}$	Low Level Output Voltage $I_O = 4.0 \text{ mA D.C.}, V_{CC} = \text{min.}$			0.45	V
$I_I$	Input Leakage Current $V_{CC} = \text{max.}, \text{GND} < V_{IN} < V_{CC}$			$\pm 10$	$\mu\text{A}$
$I_{OZ}$	Output Leakage Current $V_{CC} = \text{max.}, \text{GND} < V_{OUT} < V_{CC}$			$\pm 10$	$\mu\text{A}$
$I_{SC}^{(7)}$	Output Short Circuit Current $V_{CC} = \text{max.}, V_{OUT} = 0.5V$		20	30	mA
$I_{SB}^{(8)}$	Standby Current $V_{CC} = \text{max.}, V_{IN} = V_{CC} \text{ or GND, Standby mode}$		35	150	$\mu\text{A}$
$I_{CC}$	Power Supply Current $V_{CC} = \text{max.}, V_{IN} = V_{CC} \text{ or GND, No load, Input Freq.} = 1 \text{ MHz}$ Active mode (Turbo = Off), Device prog. as four 12-bit Ctrs.		30	45	mA

**NOTES:**

5. Absolute values with respect to device GND; all over and undershoots due to system or tester noise are included.
6.  $I_O$  at CMOS levels (3.84 V) = -2 mA
7. Not more than 1 output should be tested at a time. Duration of that test must not exceed 1 second.
8. With Turbo Bit Off, device automatically enters standby mode approximately 100 ns after last input transition.

**A.C. TESTING LOAD CIRCUIT****A.C. TESTING INPUT, OUTPUT WAVEFORM**

## CAPACITANCE

Symbol	Parameter	Min	Typ	Max	Unit	Conditions
C <sub>IN</sub>	Input Capacitance			15	pF	V <sub>IN</sub> = 0V, f = 1.0 MHz
C <sub>OUT</sub>	Output Capacitance			15	pF	V <sub>OUT</sub> = 0V, f = 1.0 MHz
C <sub>CLK</sub>	Clock Pin Capacitance			25	pF	V <sub>OUT</sub> = 0V, f = 1.0 MHz
C <sub>VPP</sub>	V <sub>PP</sub> Pin Capacitance			160	pF	CLK2, V <sub>OUT</sub> = 0V, f = 1.0 MHz

A.C. CHARACTERISTICS T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5V ±5%, Turbo Bit On<sup>(9)</sup>

Symbol	From	To	5C180-70 EP1800-2			5C180-75			5C180-90 EP1800			Non-Turbo Mode <sup>(11)</sup>	Unit
			Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
t <sub>PD1</sub>	Input <sup>(12)</sup>	Comb. Output			65			70			85	+ 30	ns
t <sub>PD2</sub>	I/O <sup>(12)</sup>	Comb. Output			70			75			90	+ 30	ns
t <sub>PD2e</sub>	I/O <sup>(13)</sup>	Comb. Output			65			70			85	+ 30	ns
t <sub>PZX</sub> <sup>(10)</sup>	I or I/O	Output Enable			70			75			90	+ 30	ns
t <sub>PXZ</sub> <sup>(10)</sup>	I or I/O	Output Disable			70			75			90	+ 30	ns
t <sub>CLR</sub>	Asynch. Reset	Q Reset			70			75			90	+ 30	ns

## NOTES:

9. Typ. Values are at T<sub>A</sub> = 25°C, V<sub>CC</sub> = 5V, Active Mode.10. t<sub>PZX</sub> and t<sub>PXZ</sub> are measured at ±0.5V from steady state voltage as driven by spec. output load. t<sub>PXZ</sub> is measured with C<sub>L</sub> = 5 pF.

11. If device is operated with Turbo Bit Off (Non-Turbo Mode) and the device has been inactive for approx. 100 ns, increase time by amount shown.

## SYNCHRONOUS CLOCK MODE A.C. CHARACTERISTICS

T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5V ±5%, Turbo Bit On<sup>(9)</sup>

Symbol	Parameter	5C180-70 EP1800-2			5C180-75			5C180-90 EP1800			Non-Turbo Mode <sup>(11)</sup>	Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
f <sub>MAX</sub>	Max Frequency 1/(t <sub>CH</sub> + t <sub>CL</sub> )—No Feedback			20.8			19.6			16.1		MHz
f <sub>CNT</sub>	Max. Count Frequency 1/t <sub>CNT</sub> —With Feedback			16.1			15.1			12.2		MHz
t <sub>SU1</sub>	Input Setup Time to Clk <sup>(12)</sup>	48			51			62			+ 30	ns
t <sub>SU2</sub>	I/O Setup Time to Clk <sup>(12)</sup>	53			56			67			+ 30	ns
t <sub>SU2e</sub>	I/O Setup Time to Clk <sup>(13)</sup>	48			51			62			+ 30	ns
t <sub>H</sub>	I or I/O Hold after Clk High	0			0			0				ns
t <sub>CO</sub>	Clk High to Output Valid			29			30			35		ns
t <sub>CNT</sub>	Register Output Feedback to Register Input— Internal Path	62			66			82			+ 30	ns
t <sub>CH</sub>	Clk High Time	24			25			30				ns
t <sub>CL</sub>	Clk Low Time	24			25			30				ns

# ASYNCHRONOUS CLOCK MODE A.C. CHARACTERISTICS

T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5V ±5%, Turbo Bit On<sup>(9)</sup>

Symbol	Parameter	5C180-70 EP1800-2			5C180-75			5C180-90 EP1800			Non-Turbo Mode <sup>(11)</sup>	Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
f <sub>AMAX</sub>	Max. Frequency 1/(t <sub>ACH</sub> + t <sub>ACL</sub> )—No Feedback			20.8			20			16.6		MHz
f <sub>ACNT</sub>	Max. Frequency 1/t <sub>ACNT</sub> —With Feedback			16.1			15.1			12.2		MHz
t <sub>ASU1</sub>	Input Setup Time to Asynch. Clock <sup>(12)</sup>	17			19			23			+ 30	ns
t <sub>ASU2</sub>	I/O Setup Time to Asynch. Clock <sup>(12)</sup>	22			25			28			+ 30	ns
t <sub>AH</sub>	Input or I/O Hold to Asynch. Clock	30			30			30				ns
t <sub>ACO</sub>	Asynch. Clk to Output Valid			70			75			90		ns
t <sub>ACNT</sub>	Register Output Feedback to Register Input— Internal Path	62			66			82			+ 30	ns
t <sub>ACH</sub>	Asynch. Clk High Time	24			25			30				ns
t <sub>ACL</sub>	Asynch. Clk Low Time	24			25			30				ns

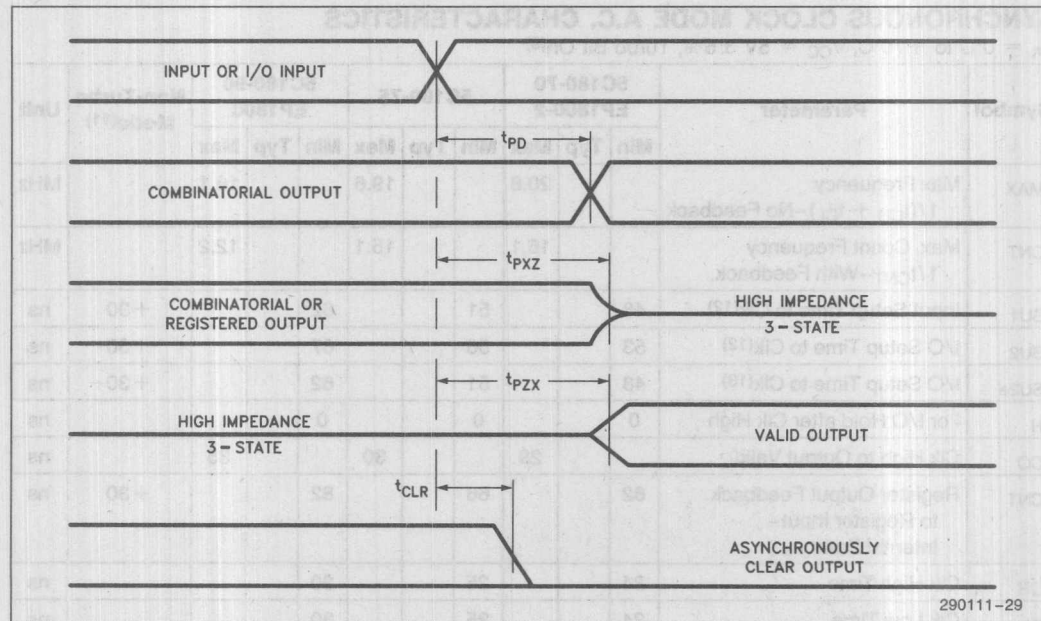
## NOTES:

12. For General and Global Macrocells.

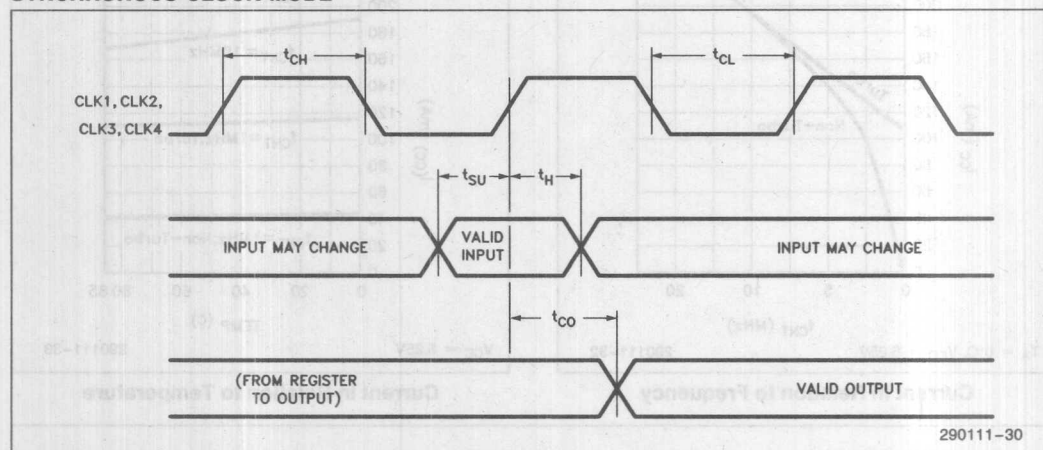
13. For Enhanced Macrocells.

## SWITCHING WAVEFORMS

### COMBINATORIAL MODE

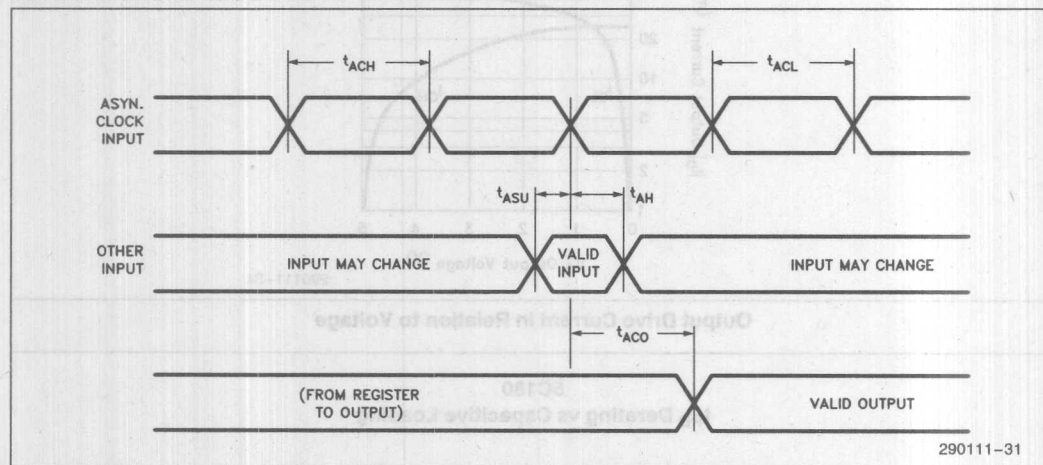


## SYNCHRONOUS CLOCK MODE

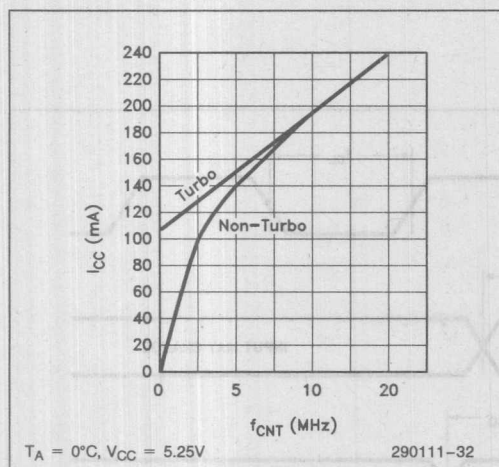


2

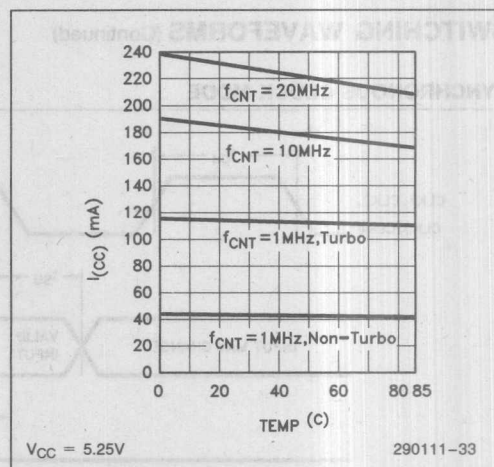
## ASYNCHRONOUS CLOCK MODE



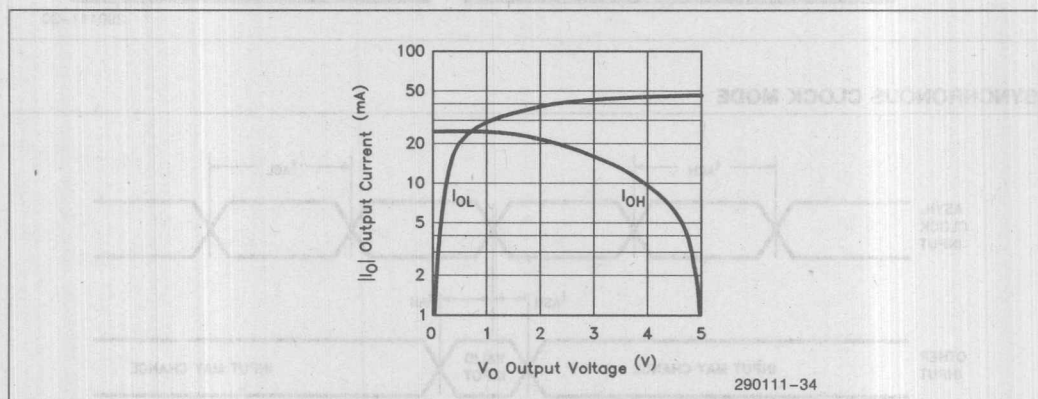




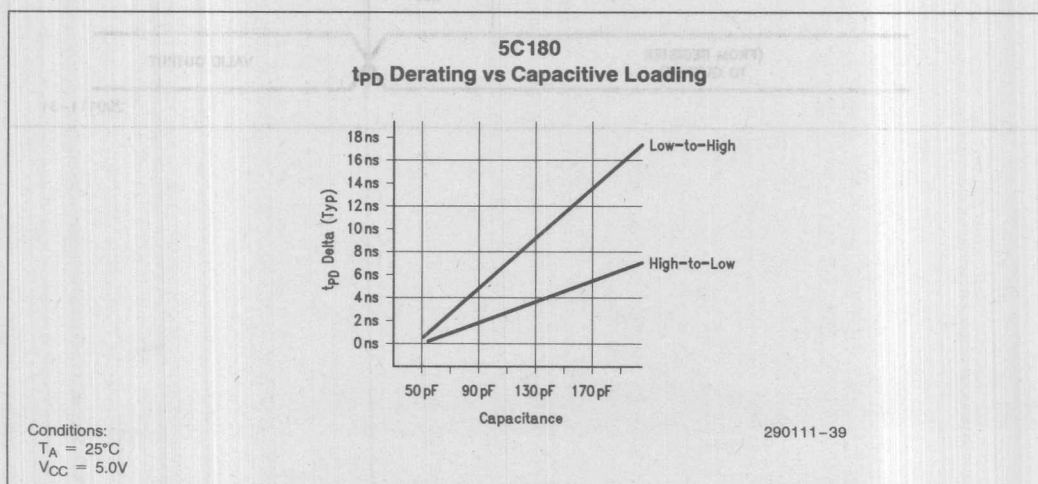
Current in Relation to Frequency



Current in Relation to Temperature

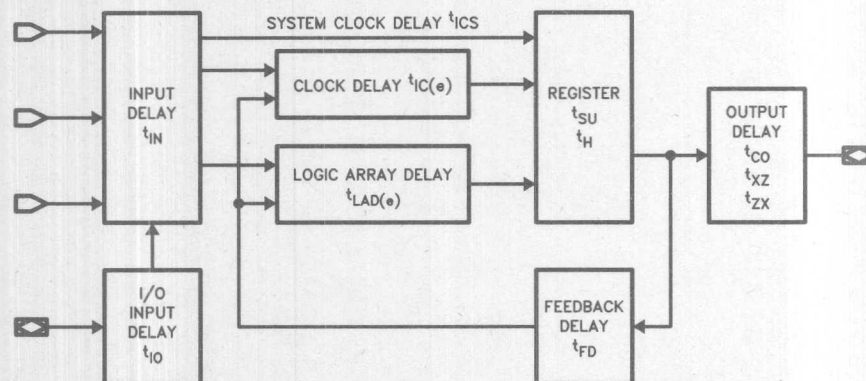


Output Drive Current in Relation to Voltage



# 5C180 INTERNAL TIMING

The following internal timing model and specifications are provided to aid in determining the different timing parameters for all permutations of timing paths through the device. The mnemonics in the table represent *internal parameters* only and should not be confused with external timing parameters shown in previous tables, even though some mnemonics are the same.



290111-38

Symbol	Parameter	5C180-70 EP1800-2		5C180-75		5C180-90 EP1800		Non-Turbo Mode(11)		Unit
		Min	Max	Min	Max	Min	Max	Min	Max	
$t_{IN}$	Input Pad and Buffer Delay		10		11		14		0	ns
$t_{IO}$	I/O Input Pad and Buffer Delay		5		5		5		0	ns
$t_{LADe}$	Enhanced Logic Array Delay		35		37		43		30	ns
$t_{LAD}$	Logic Array Delay		40		42		48		30	ns
$t_{OD}$	Output Buffer and Pad Delay		15		17		23		0	ns
$t_{ZX}$	Output Buffer Enable		15		17		23		0	ns
$t_{XZ}$	Output Buffer Disable		15		17		23		0	ns
$t_{SU}$	Register Setup Time	12		13		18		0		ns
$t_{HS}$	Register Hold Time (System Clock)	0		0		0		0		ns
$t_H$	Register Hold Time	30		30		30		0		ns
$t_{ICe}$	Enhanced Clock Delay		35		37		43		30	ns
$t_{IC}$	Clock Delay		40		42		48		30	ns
$t_{ICS}$	System Clock Delay		4		4		4		0	ns
$t_{FD}$	Feedback Delay		10		11		16		-30	ns
$t_{CLR e}$	Enhanced Register Clear Time		35		37		43		30	ns
$t_{CLR}$	Register Clear Time		40		42		48		30	ns









PAGE	CONTENTS	PAGE	CONTENTS
3-1	INTRODUCTION	3-1	INTRODUCTION
3-2	PERFORMANCE	3-2	PERFORMANCE
3-3	ARCHITECTURAL SUPERSET	3-3	ARCHITECTURAL SUPERSET
3-4	FEATURES	3-4	FEATURES
3-5	Standard PALs and Their Limitations	3-5	Standard PALs and Their Limitations
3-6	GAAs Provide a Partial Improvement	3-6	GAAs Provide a Partial Improvement
3-7	85C220/85C224—The Total Solution	3-7	85C220/85C224—The Total Solution
3-8	Other Board Design Considerations	3-8	Other Board Design Considerations
3-9	METASTABILITY CHARACTERISTICS	3-9	METASTABILITY CHARACTERISTICS
3-10	ACKNOWLEDGEMENTS	3-10	ACKNOWLEDGEMENTS

November 1992

3

# 85C220/85C224 Design Guide

**DANIEL E. SMITH, THOM BOWNS,  
AND TERRY BAUCOM**  
INTEL CORPORATION  
PROGRAMMABLE LOGIC APPLICATIONS

Order Number: 292073-003

## 85C220/85C224 Design Guide

CONTENTS	PAGE	CONTENTS	PAGE
INTRODUCTION .....	3-3	ELECTRICAL CHARACTERISTICS .....	3-7
PERFORMANCE .....	3-3	Output Slew Rates .....	3-7
ARCHITECTURAL SUPERSET		$t_{PD}$ Characteristics .....	3-8
FEATURES .....	3-4	$t_{CO1}$ Characteristics .....	3-10
Standard PALs and Their Limitations .....	3-4	$I_{OL}$ Characteristics .....	3-12
GALs Provide a Partial Improvement .....	3-4	Tracking Critical Parameters over	
85C220/85C224—The Total Solution .....	3-4	Temperature .....	3-12
Architecture Summary .....	3-7	Ground Bounce/Noise on Unswitched	
		Outputs .....	3-12
		Other Board Design Considerations .....	3-16
		METASTABILITY	
		CHARACTERISTICS .....	3-16
		ACKNOWLEDGEMENTS .....	3-16

## INTRODUCTION

In the past decade and half, system designers have come to depend on PLDs (Programmable Logic Devices) to implement random logic and interface circuits because of the low cost and high performance these devices offer. Bipolar PLDs (commonly called PALs\*) were the first class of devices to become a common part of the designer's repertoire. As power requirements and head dissipation of bipolar PLDs became a design and/or reliability limiter in the mid-1980's, designers began using CMOS PLDs (GALs\*\* and EPLDs).

In 1989 and 1990 Intel began shipping its fast 85C220 and 85C224 PLDs. These PLDs are supersets of common 20-pin and 24-pin PALs/GALs. Both the 85C220 and 85C224 exhibit extremely fast speeds, low power consumption, and high integration, which make them ideally suited for high-speed microcomputer system applications. This application note covers the architectural features that distinguish these Intel PLDs (also called  $\mu$ PLDs for Microcomputer Programmable Logic Devices) from their competition. It also covers performance, power requirements, and heat dissipation.

Finally, it describes electrical characteristics of the devices, along with guidelines for taking best advantage of device behavior while mitigating tradeoffs that may be encountered.

## PERFORMANCE

PLDs can be measured in terms of propagation delay ( $t_{PD}$ ) or state machine frequency ( $1/t_{SU} + t_{CO}$ ). In terms of propagation delay, performance of the first CMOS PLDs lagged behind bipolar PLDs by 15 ns–20 ns. This gap has been closed to 2.5 ns or less. In terms of state machine frequency, CMOS PLDs also historically lagged bipolar PLDs significantly. State-machine performance of Intel PLDs equals or surpasses bipolar PLDs. Table 1 lists the key speed parameters of 16-series and 20-series PALs and GALs, and the 85C220/85C224. The additional performance provided by the 85C220/85C224-80 can allow faster state machine designs or provide additional margin for existing designs. The 85C220/85C224-7 provide fast combinatorial logic at significantly reduced power consumption over bipolar devices.

Table 1. PAL/GAL/85C220/85C224 Performance Comparison

Parameter	16R8-D 20R8-D	16R8-E 20R8-E	16V8-10 20V8-10	85C220-7 85C224-7	85C220-80 85C224-80	85C220-100*** 85C224-100	Units
$t_{PD}$	10	7.5	10	7.5	10	7.5	ns
$t_{SU}$	10	7	10	7	7	4.5	ns
$t_{CO}$	7	6.5	8	6.5	5.5	5.5	ns
$f_{CNT1}$	58.8	74	55.5	74	80	100	MHz
$t_{CNT}$	16.5	10	Not Avail.	10	10	10	ns
$f_{CNT2}$	60	100	Not Avail.	100	100	115	MHz
$f_{MAX}$	62.5	100	62.5	100	111	115	MHz

Italics indicate fastest specification for that category.

### Parameter Definitions:

- $t_{PD}$  — Propagation Delay, Input or I/O to Output Valid Delay
- $t_{SU}$  — Input or I/O Setup Time to Clock
- $t_{CO}$  — Clock to Output Valid Delay
- $f_{CNT1}$  — Max. External Counter Frequency ( $1/t_{SU} + t_{CO}$ )
- $t_{CNT}$  — Clock to Feedback Setup for Next Clock—Internal Path
- $f_{CNT2}$  — Max. Internal Counter Frequency ( $1/t_{CNT}$ )
- $f_{MAX}$  — Max. Pipelined Frequency

\*PAL® is a registered trademark of Advanced Micro Devices.

\*\*GAL® is a registered trademark of Lattice Semiconductor, Inc.

\*\*\*85C220-100/85C224-100 is under development.



## ARCHITECTURAL SUPERSET FEATURES

This section summarizes the architectural limitations of PALs and GALs and describes the superset features of the 85C220/85C224 devices.

### Standard PALs and Their Limitations

Designers are familiar with common 16-series and 20-series PAL and GAL devices. The layout is standard with clock, output enable, inputs and outputs in the same pin locations across families of devices (all signals on R4, R6, and R8 devices; inputs and outputs on L8 devices). Seven or eight p-terms (product-terms) are available for logic implementation. The major task with PALs is to pick the device with the desired range of inputs and registers. The designer then fits the design into the device, modifying the design to fit the architecture of the device.

Architectural limitations that experienced PAL users have all encountered include:

1. One active output polarity per device (usually active low)
2. Only 7 SOP (Sum-of-Products) p-terms on combinatorial outputs
3. Unusable pin on R4, R6 and R8 devices if the dedicated OE is not used
4. No feedback on the outside macrocells (#0 and #7) of L8 devices
5. Wasted resources in cases where the number and type of outputs does not match device resources. (For example, a 16R-series or 20R-series PAL cannot implement a design requiring 5-registered and 3-combinatorial outputs. An R6 would be needed to implement the 5 registers and 2 of the 3 combinatorial outputs. One register in the R6 would not be used and the 3rd combinatorial output would have to be implemented in discrete logic or in an additional PAL.)

### GALs Provide a Partial Improvement

The GAL architecture overcomes some of these limitations. GALs offer programmable output polarity so that any output can be active high or active low. Any combination of registered/combinatorial outputs can be implemented in a single device. But several architectural limitations are still present in GALs:

1. Only 7 SOP p-terms are available for combinatorial outputs if a p-term is used for an OE signal
2. The dedicated OE pin is unusable if registers are used but the OE control is not needed

3. No feedback on the outside macrocells (#0 and #7) when the device is configured for all combinatorial outputs.

Thus while GALs provide greater flexibility than PALs, GAL users still find themselves designing around the architectural limitations.

### 85C220/85C224—The Total Solution

With the Intel 85C220/85C224 PLDs, the architectural limitations of PALs and GALs have been overcome. The 85C220 is an architectural superset of 16-series (20-pin) PALs and GALs, while the 85C224 is an architectural superset of 20-series (24-pin) devices. Let's look at the superset features.

#### SOP Invert

Figure 1 compares the macrocells (output structures) of PALs, GALs, and the 85C220/85C224. Note that the 85C220/85C224 and the GAL contain a programmable invert option to allow each output to be individually configured as active low or active high. This feature allows logic compilers to use DeMorgan's inversion techniques to fit equations into an 85C220/85C224 that would not fit into a standard PAL. Larger equations will fit into the 85C220/85C224 that will not fit in 16-series and 20-series PALs.

#### Independently Configurable Outputs

The 85C220/85C224 and the GAL allow any combination of registered and combinatorial outputs in a single device. This feature means that designers will not have to leave outputs unused as with PALs.

#### Output Enable P-Term

Figure 1 also shows that the 85C220/85C224 contains an output enable p-term in addition to the 8 SOP p-terms (9 p-terms total). L8 PALs and GALs borrow one of the SOP p-terms to implement an OE equation, leaving only 7 p-terms to implement the SOP. In cases where an SOP equation requires all 8 p-terms, the PAL/GAL designer must route part of the equation through another macrocell first (and accept the additional delay), or choose a different device. Since the 85C220/85C224 does not have this limitation, larger designs will fit more often in Intel PLDs.

#### Output Feedback

Note that the two outside macrocells (#0 and #7) on L8 PALs and GALs configured for combinatorial operation do not allow feedback to the logic array. If this feedback is needed, a different device must be chosen.

This may mean switching to an 18P8, 18CV8, etc., or moving up to a larger device with more inputs. All 8 outputs on the 85C220/85C224 can feed back a combinatorial signal from the I/O pin to the logic array. The need to change devices or move up to a larger device is diminished for 85C220/85C224 users.

### No Dedicated Output Enable Pin

When using PALs and GALs configured for registered operation, the OE pin can only be used as an OE signal.

If no OE function is needed, the pin must be tied high; it cannot be used as a standard input to the logic array. The 85C220/85C224, however, does not contain a dedicated output enable pin, making these devices more flexible than PALs and GALs. With the 85C220/85C224, if the registers do not require an output enable, the internal p-term can hold the output buffers in the enabled state. The corresponding pin can then be used as a dedicated input. If a global OE is needed, then the OE p-terms for all eight macrocells can be programmed identically.

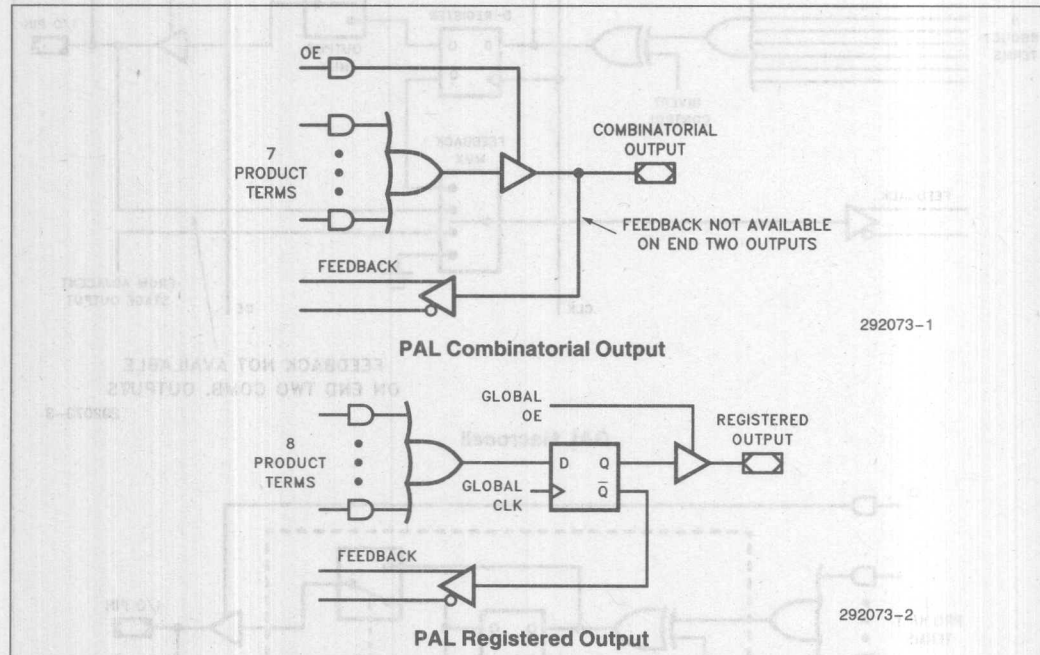


Figure 1. Output/Macrocell Architecture Comparison

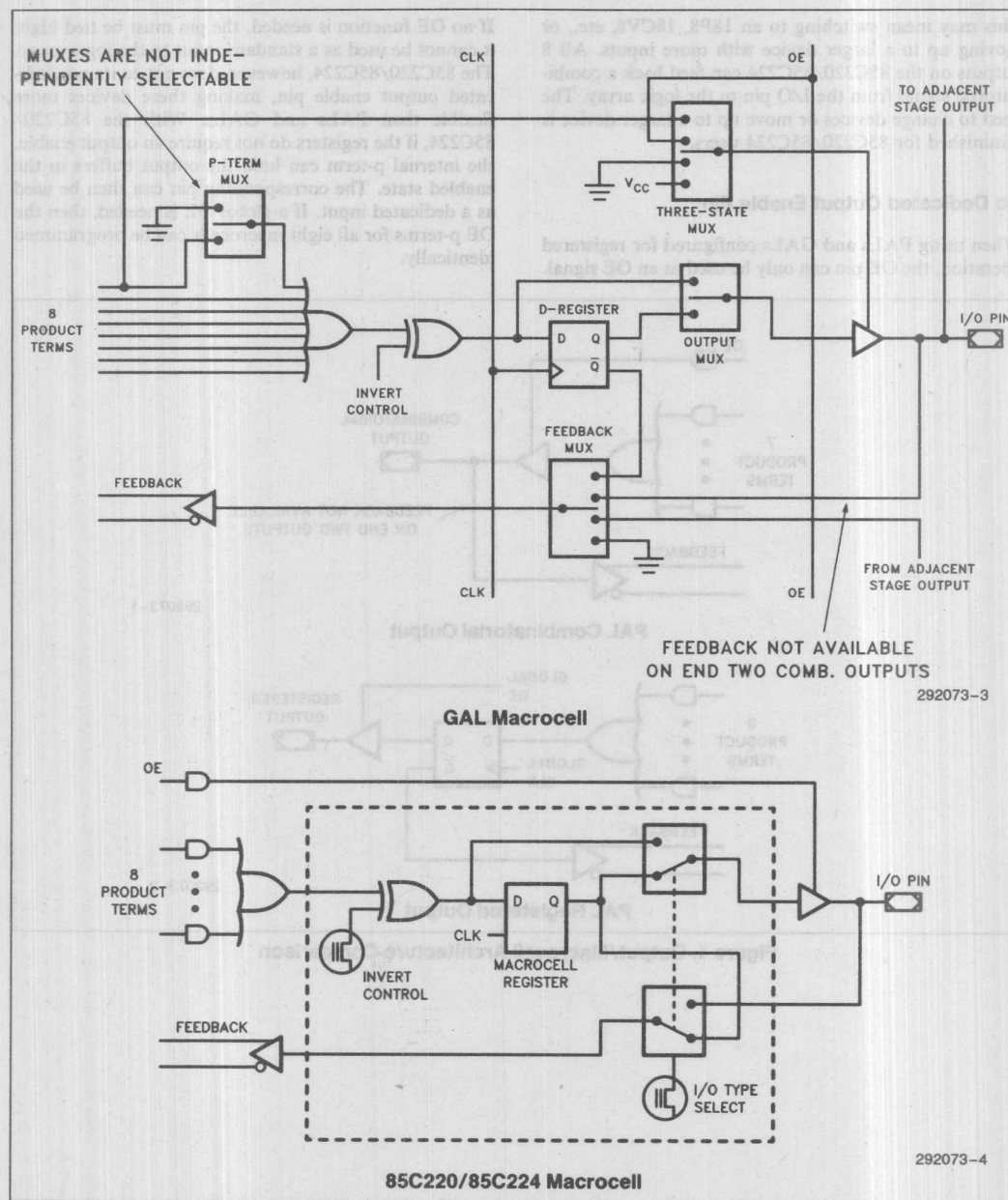


Figure 1. Output/Macrocell Architecture Comparison (Continued)

## Architecture Summary

The combination of PAL/GAL compatibility along with the superset features on the 85C220/85C224  $\mu$ PLDs allows more logic to be implemented in these devices than in standard PAL/GAL architectures. The need to stock several different PAL/GAL architectures (in multiple packages) is reduced when using the 85C220/85C224 architectures. 85C220/85C224 users will find that their designs outgrow their PLDs significantly less often than PAL/GAL users.

## ELECTRICAL CHARACTERISTICS

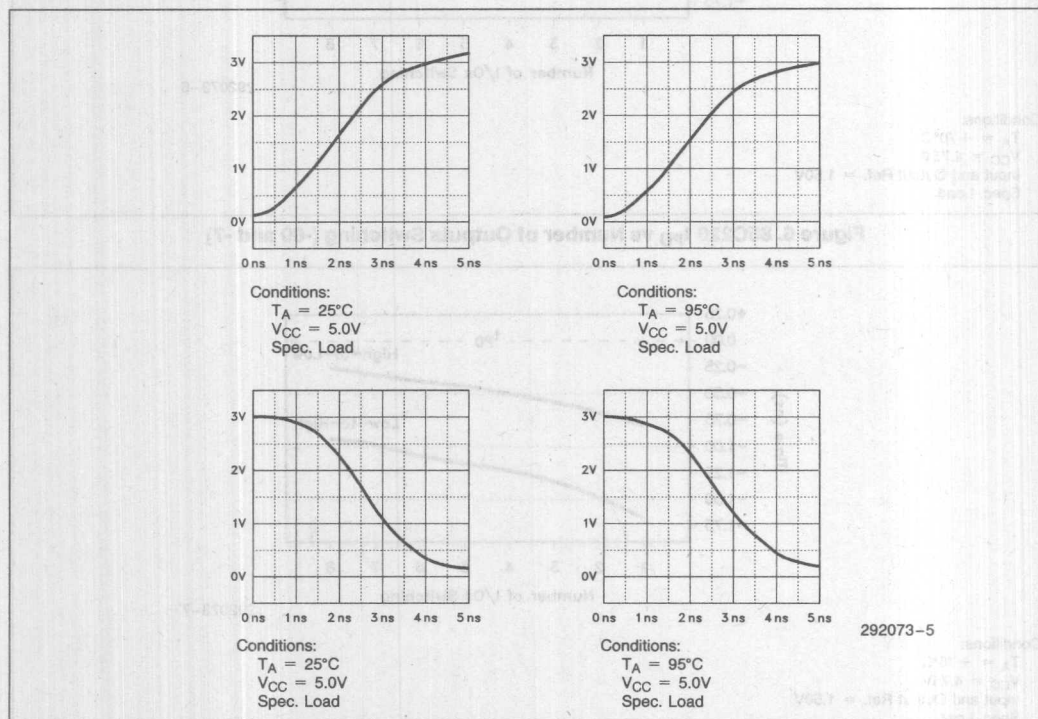
This section provides characterization data for the 85C220/85C224 that can be of great help to designer's working on high-speed systems. Information is presented in the following order:

- Output Slew Rates
- $t_{PD}$  Characteristics

- $t_{CO}$  Characteristics
- $I_{OL}$  Characteristics
- Tracking Critical Parameters over Temperature
- Ground Bounce/Noise on Unswitched Outputs

## Output Slew Rates

Output buffers for 85C220/85C224-80 devices have a nominal slew rate of 1 V/ns low-to-high and 1.5 V/ns high-to-low. At 95°C, the slew rate slows slightly to 0.8 V/ns low-to-high, and 1.2 V/ns high-to-low. This slew rate is fast enough to meet the requirements of all data sheet specifications, but slow enough to minimize problems such as ground bounce and transmission line effects. Figures 2 through 5 show slew rates as measured on 85C220/85C224-80 devices.



Figures 2-5. 85C220/85C224-80 Output Slew Rates



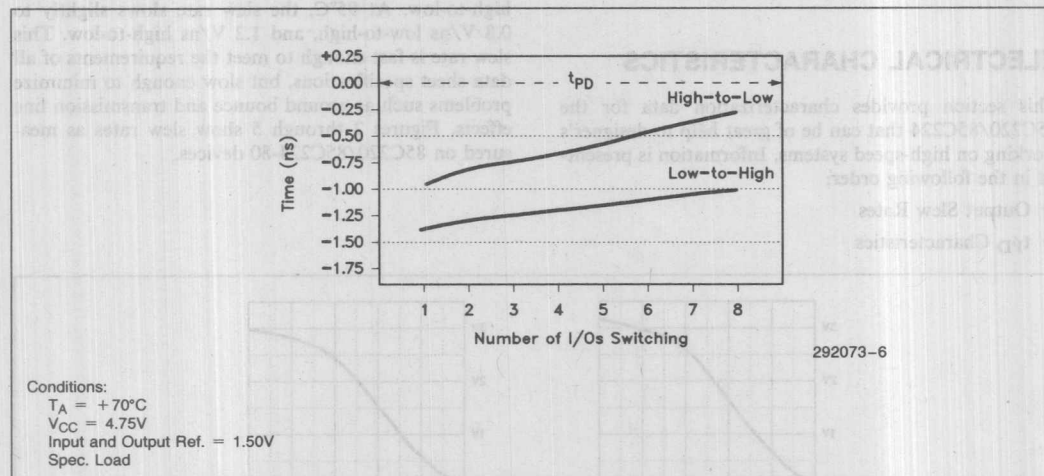
## **t<sub>PD</sub> Characteristics**

This section shows how propagation delay for the 85C220/85C224 PLDs is affected by factors that vary from one application to another.

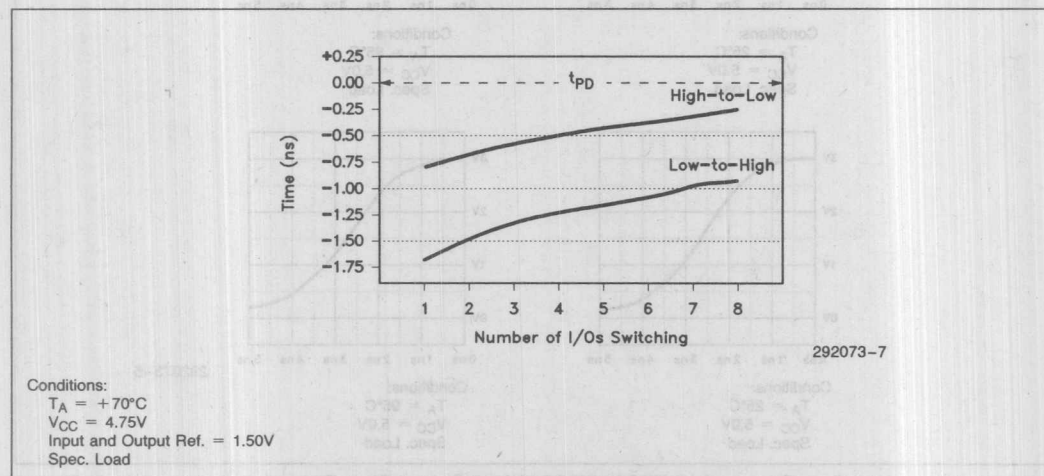
### **t<sub>PD</sub> vs Number of Outputs Switching**

As the number of device outputs switching simultaneously increases, average propagation delay through

devices also increases. This increase is related to the ability of package power and ground leads to channel the additional current. Figures 6 and 7 show the relation of t<sub>PD</sub> to the number of outputs switching for the 85C220 and 85C224, respectively. Note that this data reflects worst case values. Typical parts can be 1 ns–2 ns faster. This data helps designers estimate how much margin exists in a high-speed design.



**Figure 6. 85C220 t<sub>PD</sub> vs Number of Outputs Switching (-80 and -7)**



**Figure 7. 85C224 t<sub>PD</sub> vs Number of Outputs Switching (-80 and -7)**

### $t_{PD}$ vs $C_L$

Knowledge of how PLDs behave as capacitive loading is increased is an important consideration when designing high-speed systems. Figure 8 shows derating from specified values for a typical 85C220/85C224-80 at high temperature, low  $V_{CC}$  conditions for both low-to-high and high-to-low transitions as capacitance increases. These characteristics can help designers trade off system margin for additional delays incurred due to higher capacitive loads.

### $t_{PD}$ vs Number of P-Terms Switching

As additional p-terms for a macrocell are used, internal device loading causes propagation delay to increase or decrease slightly. Figure 9 shows this slowdown for the 85C220/85C224-80 and -7 at room temperature conditions with 2 outputs switching. Data is shown for both low-to-high and high-to-low transitions.

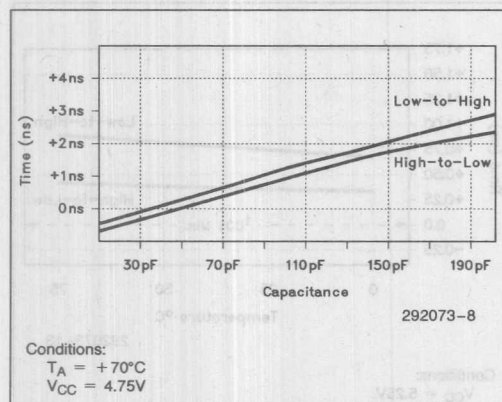


Figure 8. 85C220/85C224-80  $t_{PD}$  vs  $C_L$

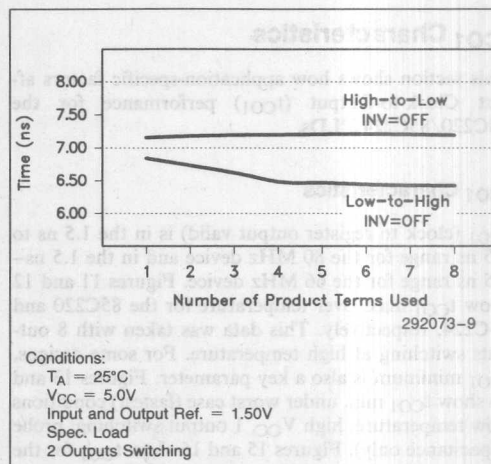


Figure 9. 85C220  $t_{PD}$  vs Number of P-Terms Switching

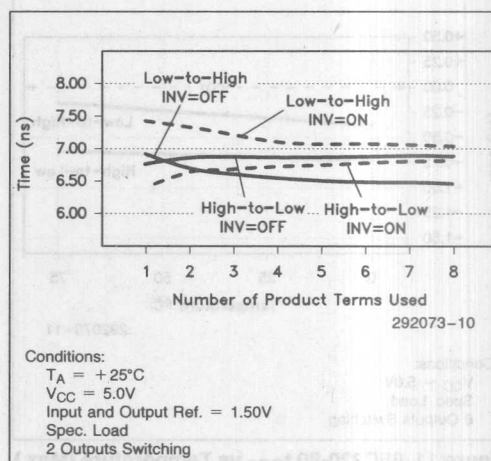


Figure 10. 85C224  $t_{PD}$  vs Number of P-Terms Switching

## $t_{CO1}$ Characteristics

This section shows how application-specific factors affect Clock-to-Output ( $t_{CO1}$ ) performance for the 85C220/85C224 PLDs.

### $t_{CO1}$ Characteristics

$t_{CO1}$  (clock to register output valid) is in the 1.5 ns to 5.5 ns range for the 80 MHz device and in the 1.5 ns–6.5 ns range for the 66 MHz device. Figures 11 and 12 show  $t_{CO1}$  max. over temperature for the 85C220 and 85C224, respectively. This data was taken with 8 outputs switching at high temperature. For some designs,  $t_{CO1}$  minimum is also a key parameter. Figures 13 and 14 show  $t_{CO1}$  min. under worst case (fastest) conditions (low temperature, high  $V_{CC}$ , 1 output switching, probe capacitance only). Figures 15 and 16 show  $t_{CO1}$  for the devices with 1 to 8 outputs switching under worst case conditions (high temperature, low  $V_{CC}$ ).

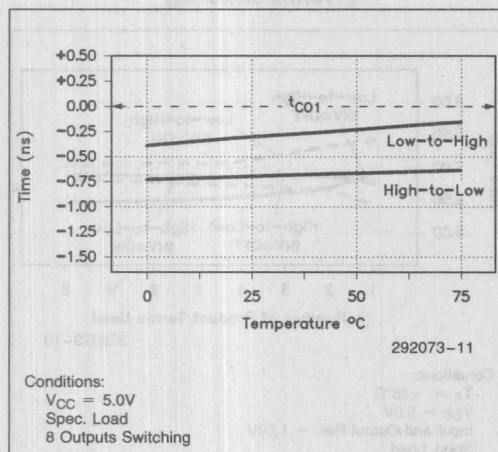


Figure 11. 85C220-80  $t_{CO1}$  vs Temperature (Max.)

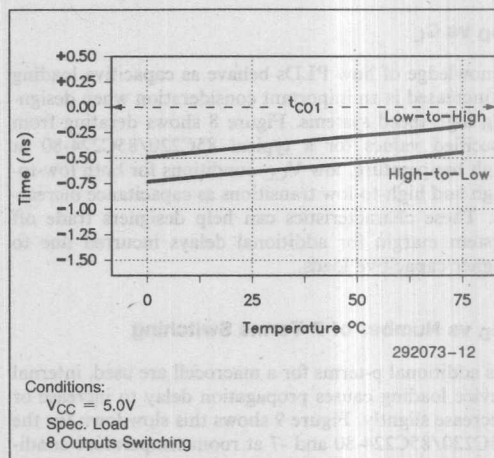


Figure 12. 85C224-80  $t_{CO1}$  vs Temperature (Max.)

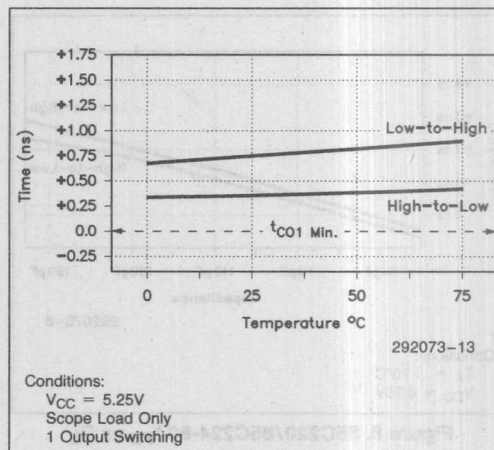
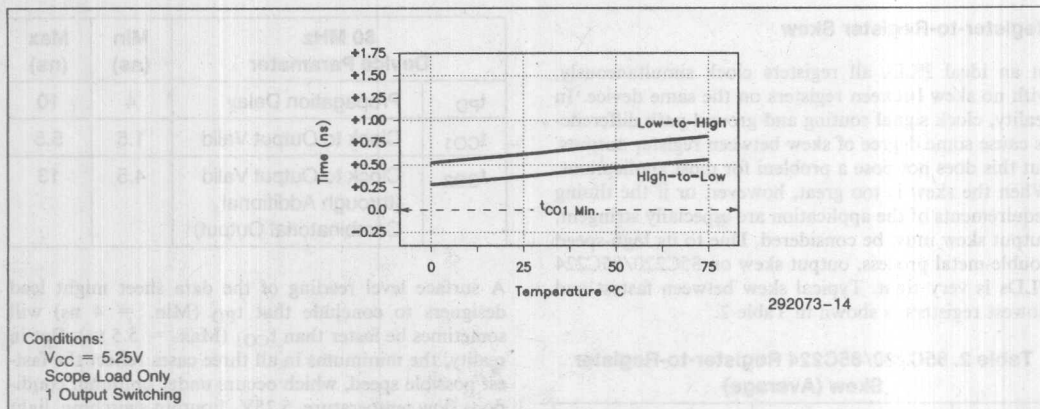
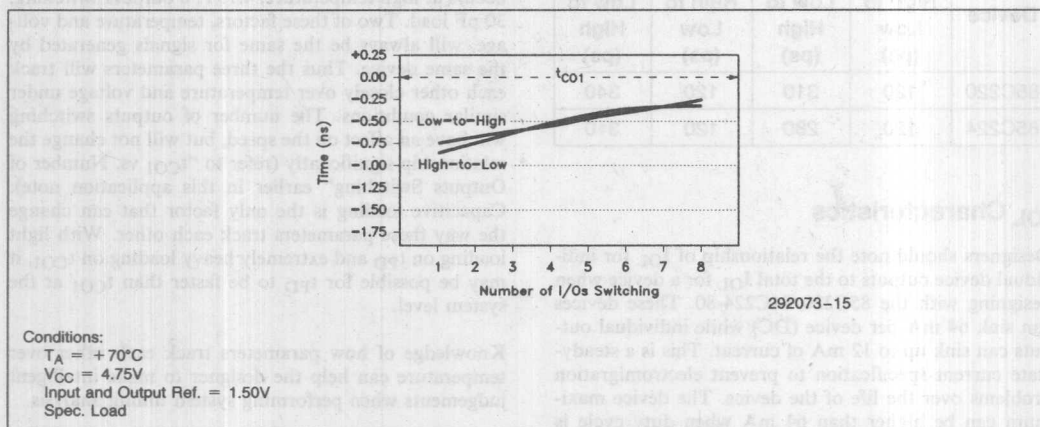
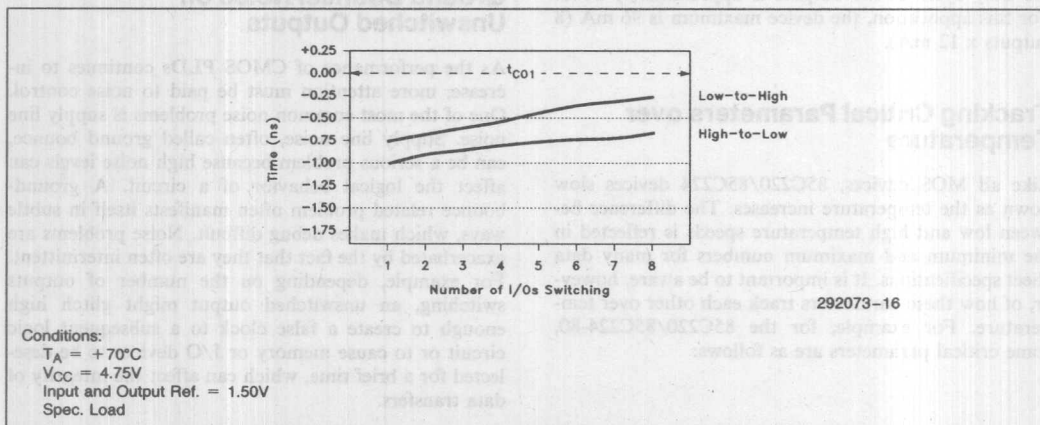


Figure 13. 85C220-80  $t_{CO1}$  vs Temperature (Min.)

Figure 14. 85C224-80  $t_{CO1}$  vs Temperature (Min.)Figure 15. 85C220-80  $t_{CO1}$  vs Number of Outputs SwitchingFigure 16. 85C224-80  $t_{CO1}$  vs Number of Outputs Switching



### Register-to-Register Skew

In an ideal PLD, all registers clock simultaneously, with no skew between registers on the same device. In reality, clock signal routing and ground path differences cause some degree of skew between register outputs, but this does not pose a problem for most applications. When the skew is too great, however, or if the timing requirements of the application are especially stringent, output skew must be considered. Due to its high-speed double-metal process, output skew on 85C220/85C224 PLDs is very tight. Typical skew between fastest and slowest registers is shown in Table 2.

**Table 2. 85C220/85C224 Register-to-Register Skew (Average)**

Device	0°C		+70°C	
	High to Low (ps)	Low to High (ps)	High to Low (ps)	Low to High (ps)
85C220	120	310	120	340
85C224	120	280	120	310

### I<sub>OL</sub> Characteristics

Designers should note the relationship of I<sub>OL</sub> for individual device outputs to the total I<sub>OL</sub> for a device when designing with the 85C220/84C224-80. These devices can sink 64 mA per device (DC) while individual outputs can sink up to 12 mA of current. This is a steady-state current specification to prevent electromigration problems over the life of the device. The device maximum can be higher than 64 mA when duty cycle is taken into account. For example, in an 8-bit counter the duty cycle of the outputs is approximately 50%. For this application, the device maximum is 96 mA (8 outputs x 12 mA).

### Tracking Critical Parameters over Temperature

Like all MOS devices, 85C220/85C224 devices slow down as the temperature increases. The difference between low and high temperature speeds is reflected in the minimum and maximum numbers for many data sheet specifications. It is important to be aware, however, of how these parameters track each other over temperature. For example, for the 85C220/85C224-80, some critical parameters are as follows:

80 MHz Device Parameter		Min (ns)	Max (ns)
t <sub>PD</sub>	Propagation Delay	4	10
t <sub>CO1</sub>	Clock to Output Valid	1.5	5.5
t <sub>CO2</sub>	Clock to Output Valid (through Additional Combinatorial Output)	4.5	13

A surface level reading of the data sheet might lead designers to conclude that t<sub>PD</sub> (Min. = 4 ns) will sometimes be faster than t<sub>CO1</sub> (Max. = 5.5 ns). But in reality, the minimums in all three cases reflect the fastest possible speed, which occurs under optimum conditions (low temperature, 5.25V, 1 output switching, light load). The maximums reflect the slowest speed, which occurs at high temperature, 4.75V, 8 outputs switching, 30 pF load. Two of these factors, temperature and voltage, will always be the same for signals generated by the same device. Thus the three parameters will track each other closely over temperature and voltage under similar conditions. The number of outputs switching will have an affect on the speed, but will not change the relationship significantly (refer to "t<sub>CO1</sub> vs. Number of Outputs Switching" earlier in this application note). Capacitive loading is the only factor that can change the way these parameters track each other. With light loading on t<sub>PD</sub> and extremely heavy loading on t<sub>CO1</sub>, it may be possible for t<sub>PD</sub> to be faster than t<sub>CO1</sub> at the system level.

Knowledge of how parameters track each other over temperature can help the designer to make intelligent judgements when performing system timing analysis.

### Ground Bounce/Noise on Unswitched Outputs

As the performance of CMOS PLDs continues to increase, more attention must be paid to noise control. One of the most common noise problems is supply line noise. Supply line noise, often called ground bounce, can be a serious problem because high noise levels can affect the logical behavior of a circuit. A ground-bounce related problem often manifests itself in subtle ways, which makes debug difficult. Noise problems are exacerbated by the fact that they are often intermittent. For example, depending on the number of outputs switching, an unswitched output might glitch high enough to create a false clock to a subsequent logic circuit or to cause memory or I/O devices to be deselected for a brief time, which can affect the integrity of data transfers.

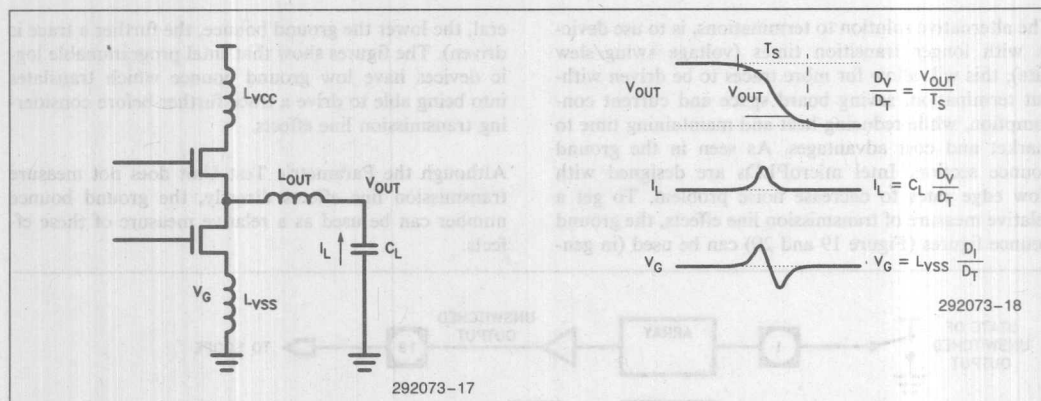


Figure 17. Basic Cause of Ground Bounce

### Basic Cause of Ground Bounce

The fundamental cause of ground bounce is switching loaded output pins very quickly. Outputs are normally capacitively loaded, and there is always intrinsic inductance in device bond wires and package lead frames. The resulting structure is a classic LC circuit, as shown in Figure 17. Charging or discharging the circuit creates a damped oscillation. The magnitude, frequency, and duration of the oscillation are determined by the component values that make up the circuit.

$V_{OUT}$  in Figure 17 is normally between 3V and 5V. Switching time ( $T_S$ ) is in the range of 2 ns–4 ns. Load capacitance ( $C_L$ ) is in the 30 pF–100 pF range. Inductance depends on the package design and materials, and is typically in the 7 nH–25 nH range. Based on these conditions, it is easy to see that ground bounce ( $V_G$ ) can be a problem. Ground bounce further increases as more outputs switch simultaneously. In actual practice, the noise levels are less than the simple analysis shown in Figure 17 suggests. This is due to the self limiting nature of the output drivers. As the ground noise causes the local (internal) ground to rise, the drive of the device is reduced, which in turn reduces  $d_i/d_t$ .

Figure 18 shows the test setup for measurements. Measurements were made on 85C220-80 devices with 7 outputs switching simultaneously and the remaining output held high or low. The noise generated on the static output by the 7 outputs switching was then measured and recorded (worst case noise condition for a static output). All outputs have 22 pF of capacitance (> 30 pF with scope probe), a 220Ω resistor to ground, and 330Ω resistor to  $V_{CC}$ . Decoupling is implemented with a 4.7 μF and a multi-layer 0.1 μF (ceramic) capacitor.

For more information on measuring ground bounce, see AP-354, Guide to the PLD Parametric Test Unit.

Figure 19 shows waveforms for the devices with 7 outputs switching. The unswitched outputs are held low.

Waveforms are shown for the 85C220-80 (10 ns), 16V8A-10, 16L8-10, and 16L8-7 in Plastic DIP. An 85C220-80 and 16V8A-10 PLCC comparison is shown in Figure 20. Table 3 shows the relative inductance of the different packages.

The values shown represent a lumped load. In an actual system, the distributed load will reduce the values slightly.

Table 3. Package vs. Typical Inductance

Package Type	Typical Inductance
Plastic DIP (with Copper Leadframe)	11 nH
PLCC	6 nH

### TRANSMISSION LINE EFFECTS/ NOISE ON SWITCHED OUTPUTS

Although a thorough discussion of transmission line effects is beyond the scope of the application note, a few key points will be discussed. Transmission line effects, like ground bounce, is a slow rate related noise issue. While ground bounce is noise on unswitched outputs, transmission line effects is noise on switched outputs.

In order to minimize transmission line effects, an engineer could use common termination techniques described in many system design guides. Adding terminations does have drawbacks of:

- Increased board space
- Increased cost
- Increased current consumption
- Increased heat

The alternative solution to terminations, is to use devices with longer transition times (voltage swing/slew rate); this will allow for more traces to be driven without termination, saving board space and current consumption, while reducing heat and maintaining time to market and cost advantages. As seen in the ground bounce section, Intel microPLDs are designed with slow edge rates to decrease noise problem. To get a relative measure of transmission line effects, the ground bounce figures (Figure 19 and 20) can be used (in gen-

eral, the lower the ground bounce, the further a trace is driven). The figures show that Intel programmable logic devices have low ground bounce which translates into being able to drive a trace further before considering transmission line effects.

Although the Parametric Test Unit does not measure transmission line effects directly, the ground bounce number can be used as a relative measure of these effects.

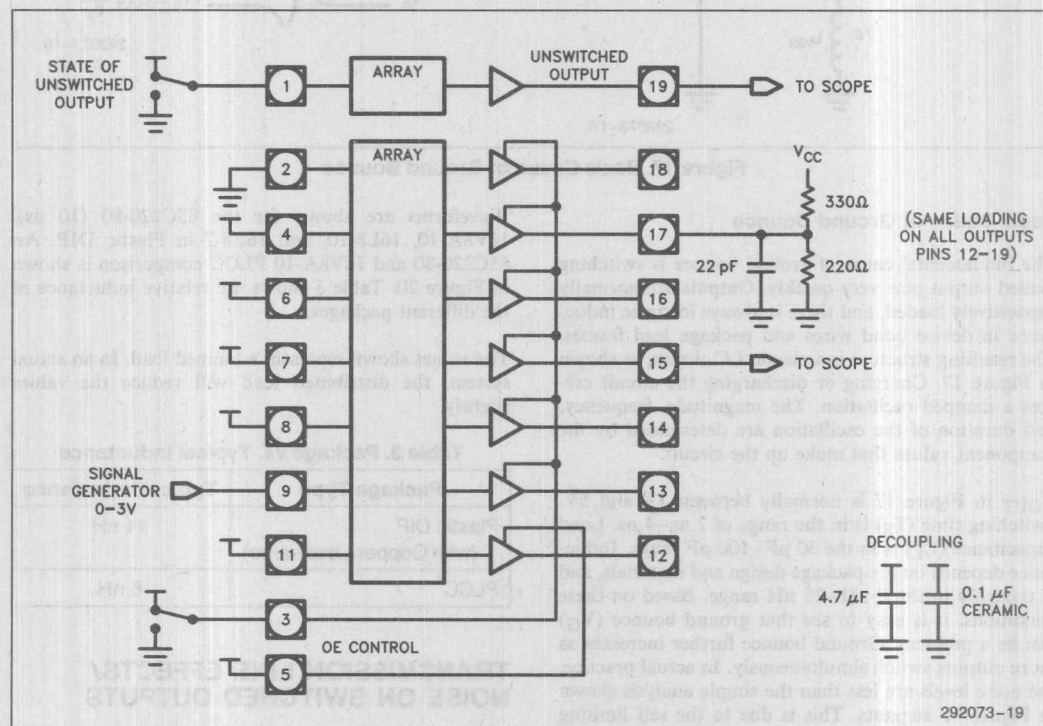
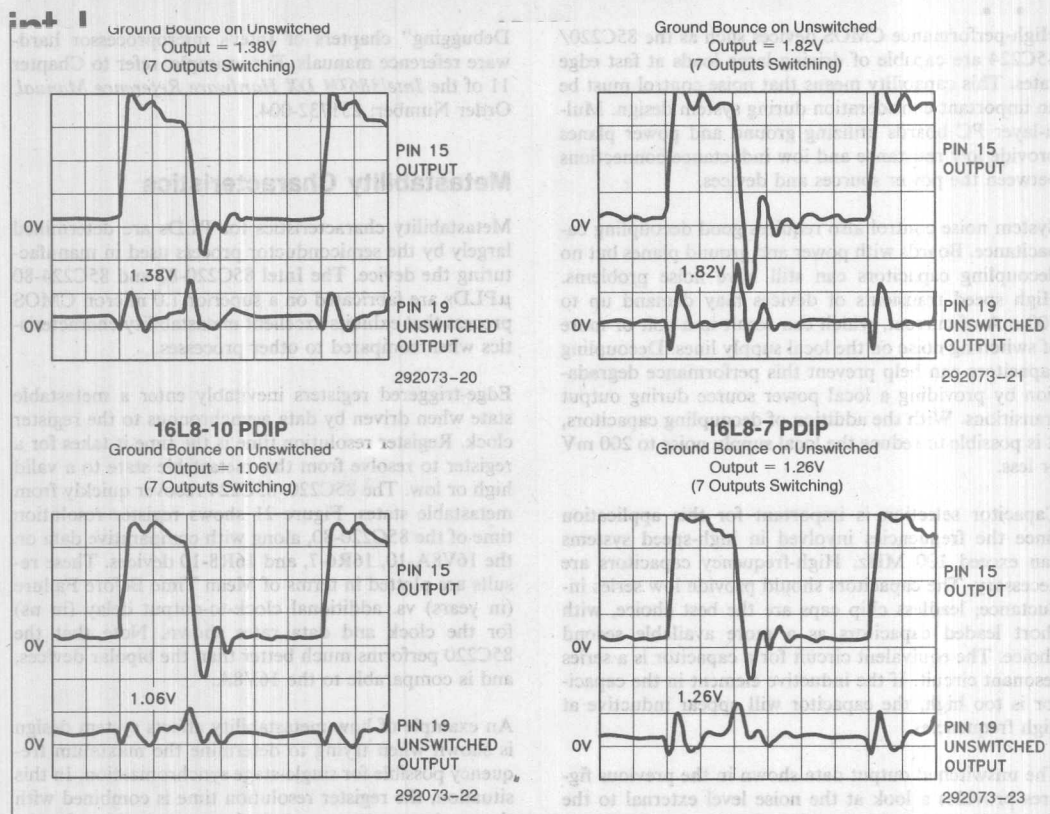
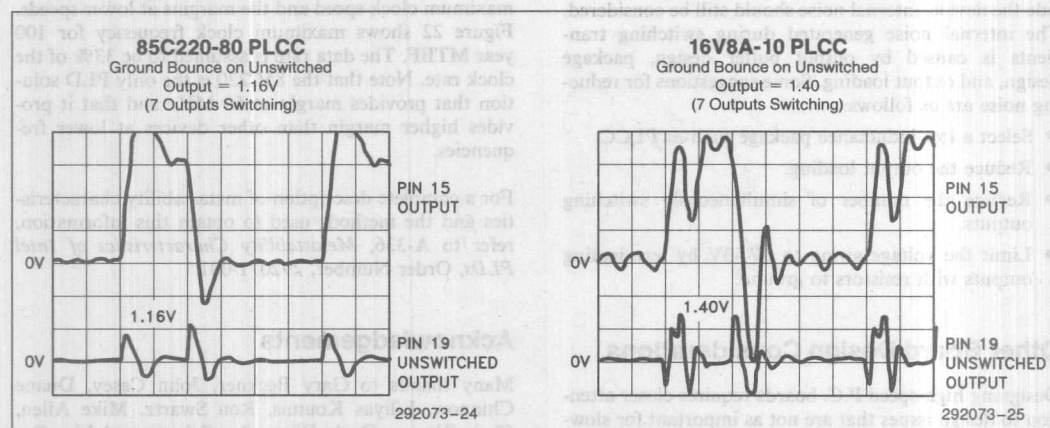


Figure 18. Ground Bounce Test Circuit



**Figure 19. Ground Bounce (PDIP)—7 Outputs Switching**



**Figure 20. Ground Bounce (PLCC)—7 Outputs Switching**



## Design Considerations

High-performance CMOS devices such as the 85C220/85C224 are capable of driving large loads at fast edge rates. This capability means that noise control must be an important consideration during system design. Multi-layer PC boards utilizing ground and power planes provide low resistance and low inductance connections between the power sources and devices.

System noise control also requires good decoupling capacitance. Boards with power and ground planes but no decoupling capacitors can still have noise problems. High speed transients of devices may demand up to 500 mA of current, which can result in a volt or more of switching noise on the local supply lines. Decoupling capacitors can help prevent this performance degradation by providing a local power source during output transitions. With the addition of decoupling capacitors, it is possible to reduce the local supply noise to 200 mV or less.

Capacitor selection is important for this application since the frequencies involved in high-speed systems can exceed 100 MHz. High-frequency capacitors are necessary. The capacitors should provide low series inductance; leadless chip caps are the best choice, with short leaded capacitors as a more available second choice. The equivalent circuit for a capacitor is a series resonant circuit. If the inductive element in the capacitor is too high, the capacitor will appear inductive at high frequencies.

The unswitched output data shown in the previous figures provides a look at the noise level external to the device. Assuming that intelligent design practices have been followed to manage noise on the supply lines outside the device, internal noise should still be considered. The internal noise generated during switching transients is caused by output buffer design, package design, and output loading. Some suggestions for reducing noise are as follows:

- Select a low-inductance package such as PLCC.
- Reduce the output loading.
- Reduce the number of simultaneously switching outputs.
- Limit the voltage swing to 0V–3V by terminating outputs with resistors to ground.

## Other Board Design Considerations

Designing high-speed P.C. boards requires closer attention to design issues that are not as important for slower systems. These elements include:

- Termination of transmission lines
- Clock signal routing
- Power distribution and heat dissipation

These topics exceed the scope of the application note. A discussion of them can be found in the "Design and Debugging" chapters of Intel's microprocessor hardware reference manuals. For example, refer to Chapter 11 of the *Intel386™ DX Hardware Reference Manual*, Order Number: 231732-004.

## Metastability Characteristics

Metastability characteristics for PLDs are determined largely by the semiconductor process used in manufacturing the device. The Intel 85C220-80 and 85C224-80  $\mu$ PLDs are fabricated on a superior 1.0 micron CMOS process that exhibits excellent metastability characteristics when compared to other processes.

Edge-triggered registers inevitably enter a metastable state when driven by data asynchronous to the register clock. Register resolution time is the time it takes for a register to resolve from this metastable state to a valid high or low. The 85C220/85C224 recover quickly from metastable states. Figure 21 shows register resolution time of the 85C220-80, along with comparative data on the 16V8A-10, 16R6-7, and 16R8-10 devices. These results are plotted in terms of Mean Time Before Failure (in years) vs. additional clock-to-output delay (in ns) for the clock and data rates shown. Note that the 85C220 performs much better than the bipolar devices, and is comparable to the 16V8A.

An example of how metastability affects system design is shown when trying to determine the maximum frequency possible for single-stage synchronization. In this situation, the register resolution time is combined with the maximum set-up and clock-to-output times for the device. The total is then used to determine both the maximum clock speed and the margins at lower speeds. Figure 22 shows maximum clock frequency for 100 year MTBF. The data rate is assumed to be 33% of the clock rate. Note that the 85C220 is the only PLD solution that provides margin at 50 MHz and that it provides higher margin than other devices at lower frequencies.

For a complete description of metastability characteristics and the methods used to obtain this information, refer to A-336, *Metastability Characteristics of Intel PLDs*, Order Number, 292071-001.

## Acknowledgements

Many thanks to Gary Beckner, John Casey, Duane Chinnow, Liliyas Koumis, Ron Swartz, Mike Allen, Chris Wawro, Darla Wirtz, Joe Salmon and Lisa Endres, for providing technical data and/or direction for this application note.

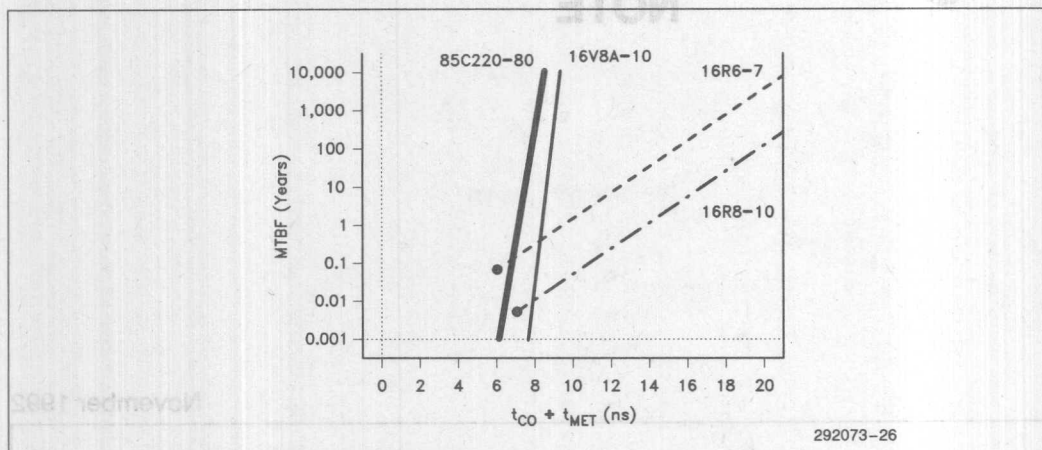


Figure 21. 85C220-80 Clock-to-Output Resolution vs MTBF

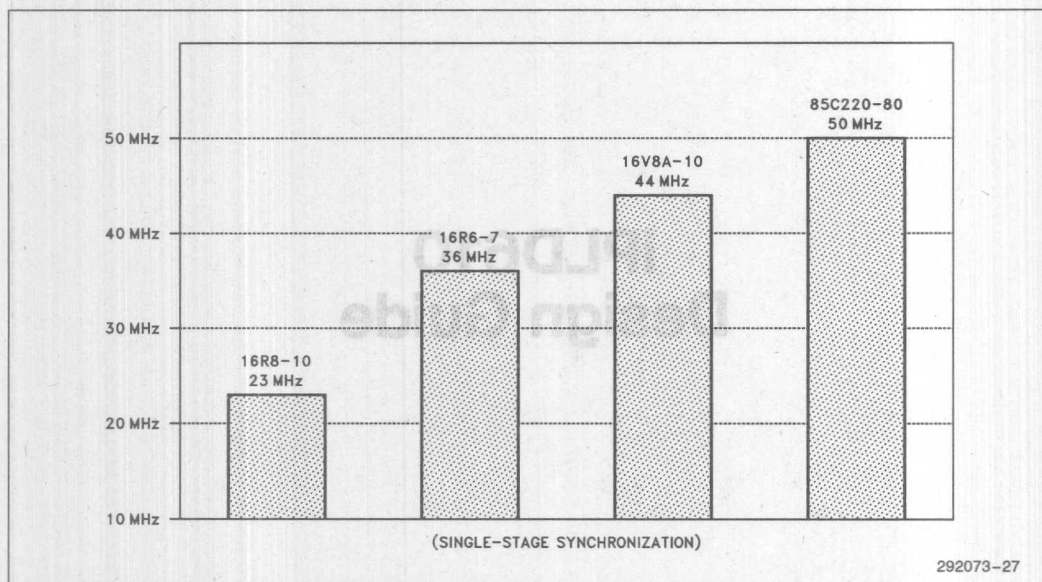
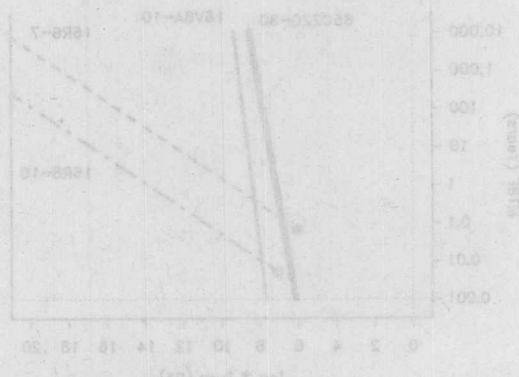


Figure 22. Maximum Frequency for 100 Year MTBF

3



November 1992

Figure 21. 86C320-60 Clock-to-Output Resolution vs MTBF

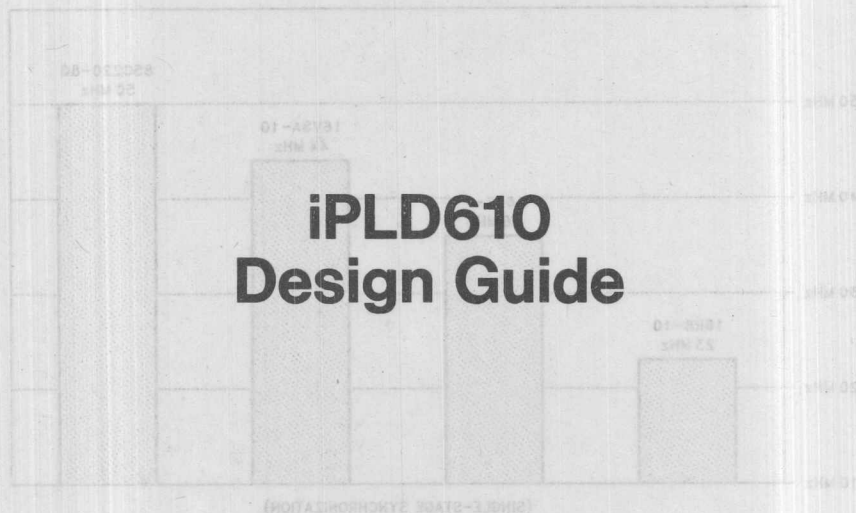


Figure 22. Maximum Frequency for 100 Year MTBF

## iPLD610 Design Guide

**JOHN CASEY**  
PROGRAMMABLE LOGIC APPLICATIONS  
INTEL CORPORATION

## CONTENTS PAGE

<b>1.0 INTRODUCTION</b> .....	3-20
Document Overview .....	3-20
Related Documents .....	3-20
<b>2.0 PRODUCT OVERVIEW</b> .....	3-20
Features .....	3-20
Packaging .....	3-21
Process .....	3-21
Device Architecture .....	3-21
Macrocell Architecture .....	3-23
<b>3.0 SPECIFICATION ANALYSIS</b> .....	3-25
3.1 D.C. Characteristics Analysis .....	3-25
TTL Compatibility .....	3-25
I <sub>OL</sub> Specifications .....	3-25
I <sub>CC</sub> Specifications .....	3-25
3.2 D.C. Specification Comparison .....	3-26
3.3 A.C. Specification Analysis .....	3-27
Combinatorial .....	3-27
Synchronous .....	3-27
Asynchronous .....	3-28
3.4 A.C. Specification Comparison .....	3-30
High Performance Devices .....	3-30
Mid-Range Performance Devices .....	3-31
PAL Comparison .....	3-32

## CONTENTS PAGE

<b>4.0 ADVANCED DESIGN ISSUES</b> .....	3-32
Output Slew Rates .....	3-32
Combinational Logic Performance (t <sub>PD</sub> ) .....	3-34
t <sub>PD</sub> vs Number of Outputs Switching .....	3-34
t <sub>PD</sub> vs C <sub>L</sub> .....	3-35
t <sub>PD</sub> vs Number of P-Terms Programmed .....	3-36
Synchronous Register Characteristics .....	3-36
t <sub>CO1</sub> vs Temperature .....	3-36
Register-to-Register Skew .....	3-37
Asynchronous Register Operation Characteristics .....	3-37
t <sub>ACO1</sub> Characteristics .....	3-37
t <sub>ASU</sub> Characteristics .....	3-38
Output Current Characteristics .....	3-38
I <sub>OL</sub> Capabilities .....	3-38
Output Drive Current .....	3-38
Other Design Considerations .....	3-39
<b>5.0 APPLICATION IDEAS</b> .....	3-40
5.1 Intel 386™ Bus State Tracker .....	3-40
5.2 Shared Memory Arbitration/Bus Control .....	3-48
5.3 High-Speed Custom Control/Status Register .....	3-54
<b>6.0 PROGRAMMING/DEVELOPMENT SUPPORT</b> .....	3-60
<b>7.0 UPGRADING TO THE iPLD610</b> .....	3-61
Upgrade From The Intel 5C060 .....	3-61
Upgrade From The Altera EP6x0 .....	3-61
Upgrade From The 22V10 .....	3-61
Architecture .....	3-62
Upgrade From The 20RA10 .....	3-65
<b>8.0 SUMMARY</b> .....	3-66



## 1.0 INTRODUCTION

Designers today are faced with the challenge of implementing very high performance systems in a cost-effective and timely manner. When these responsibilities are factored with concerns about reliability, power consumption, and migrating the design to the next performance level, the designer's job becomes more complex. When all these requirements are compared against the possible solutions spanning full-custom, semi-custom, application-specific, and programmable logic, the only sure result is a headache.

Intel's Microcomputer Programmable Logic Devices ( $\mu$ PLDs) can meet many of today's system requirements, as well as map the path to the next generation of products. Intel  $\mu$ PLDs provide a high-speed CMOS logic solution required by current microprocessors and VLSI peripherals.

### DOCUMENT OVERVIEW

This design guide provides technical support for designers, design managers, components engineers, and others interested in using Intel's iPLD610. The information provided here is intended to support both the decision making process prior to the design and, the qualification process which occurs during and after the design. The format of this design guide is as follows:

**Section 2—Product Overview:** This section discusses the device highlights and architecture of the iPLD610.

**Section 3—Specification Analysis:** The key D.C. and A.C. Specs (from the data sheet) are discussed and compared against competitive devices. This section provides a baseline for comparison and device selection. Also, some insights are provided on how to best use the data sheet specifications.

**Section 4—Advanced Design Issues:** This section includes presentation of data and discussion of issues affecting high-speed systems designs. Topics include output slew rates, effects of capacitive loading of outputs, and synchronous/asynchronous register operation.

**Section 5—Application Ideas:** This section offers some ideas on how to use the unique performance/architecture combination offered by the Intel iPLD610.

**Section 6—Programming/Development Support:** This section offers details on existing Intel development/programming tools as well as third-party support. Also discussed is programming compatibility with other devices.

**Section 7—Design Upgrade:** There may be a desire to upgrade an existing design to the Intel iPLD610. This task is discussed, specifically vis-a-vis the Intel 5C060, Altera/TI/EP6x0, 22V10, and 20RA10.

### Related Documents

Title	Intel Order No
iPLD610 CHMOS $\mu$ PLD Data Sheet	290455
iPLD910 CHMOS $\mu$ PLD Data Sheet	290456
Intel Programmable Logic Handbook	296083
Metastability Characteristics of Intel EPLDs	292071
PLD Quality and Reliability Data Summary	293003

## 2.0 PRODUCT OVERVIEW

In programmable logic, several architectures have arisen as industry standards. Each architecture has a unique set of features. The Intel iPLD610 represents the top performer in one of these industry standard architectures. The iPLD610 is an upgrade of the existing Intel 5C060 and Altera EP600 devices. Although the architectures (i.e., pin-out, logic array, macrocell features, JEDEC map) are *exactly* the same, the iPLD610 represents a very significant performance upgrade. It should also be noted that devices with this same architecture are manufactured by other leading programmable logic vendors including AMD and TI.

### FEATURES

The features provided by the Intel iPLD610 include the following highlights:

- 16 programmable macrocells (I/O pins)
- 28-pin PLCC package
- EPROM cell, CMOS Technology (UV Erasable)
- 100% Silicon Testability
- Programmable Standby Current Mode ( $< 100 \mu\text{A}$ )
- Low Operating Power (105 mA Max)
- High Performance Operation (10 ns  $t_{PD}$ , 6.5 ns  $t_{CO1}$ , 85 MHz State Machine Frequency)
- Programmable Security Bit
- Programmable Register Type

## PACKAGING

Figure 1 shows the pinouts of the DIP and PLCC packages for the iPLD. The DIP version is available in plastic, one-time-programmable (OTP). The PLCC version comes in a 28-pin OTP package.

## PROCESS

The iPLD610 uses CHMOS EPROM (floating gate) cells as logic control elements instead of fuses. The CHMOS EPROM technology reduces power consumption in comparison to bipolar devices without sacrificing speed performance. In addition, Intel's advanced CHMOS III-E EPROM process technology enables higher logic densities to be achieved with superior

speed and low-power performance over other comparable devices. Intel's  $\mu$ PLDs add the benefits of "zero" stand-by power not available on other programmable logic devices. EPROM technology allows these devices to be 100% factory tested by programming and erasing all the EPROM logic control elements.

## DEVICE ARCHITECTURE

The overall device architecture, as presented in Figure 2, shows several architectural highlights of this device. In this 24/28 pin device there are 16 macrocells, each of which represents an I/O pin, buried register or dedicated input. There are also four dedicated input pins and two dedicated clock pins.

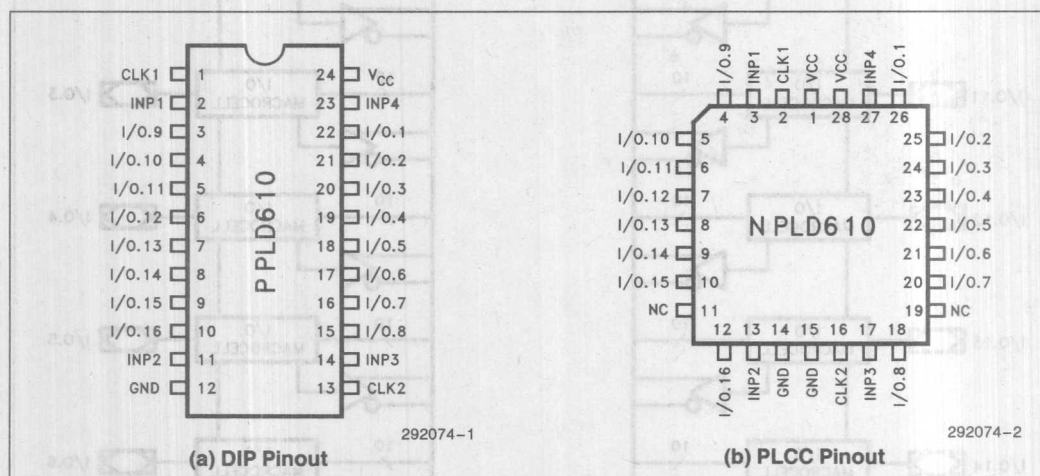


Figure 1. iPLD610 Pinout Diagrams

\*CHMOS is a patented process of Intel Corporation.

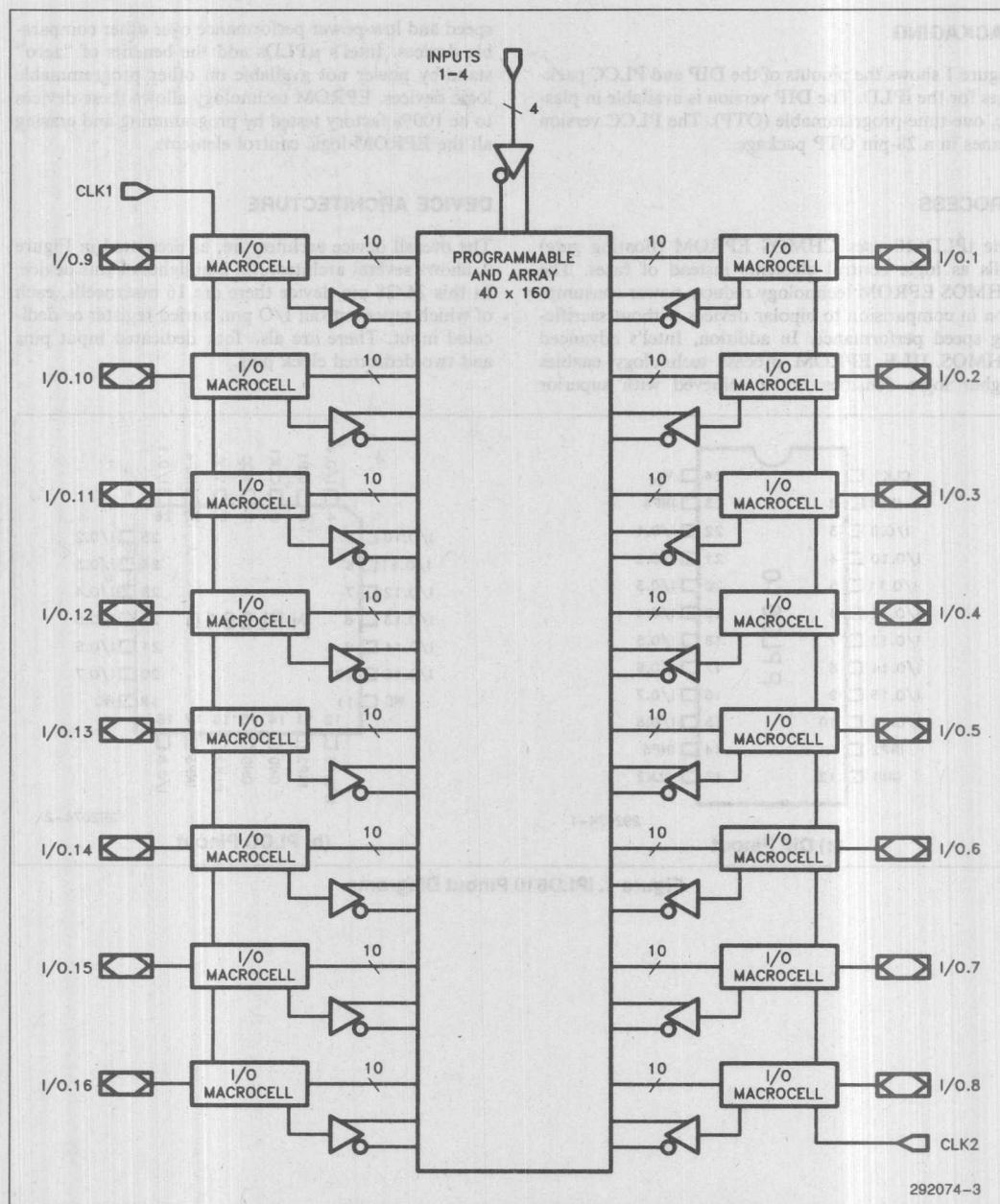


Figure 2. iPLD610 Global Architecture

The dedicated clock pins offer the system designer added flexibility. The CLK1 pin provides a synchronous clock input to macrocells 9–16 and the CLK2 pin provides a dedicated synchronous clock input for macrocells 1–8. The advantages of this architecture feature are quite obvious as the list of applications could include:

- Dual state-machine operation within the same device
- Using logic that requires operations at  $\text{CLK}_x$  and  $\overline{\text{CLK}}_x$
- Implementing system control logic requiring both a 1x CLK and a 2x CLK (as can often be the case with the Intel 80386DX/SX and 80286 microprocessors)
- Use of some macrocells as “input latches” accepting asynchronous inputs, whose states are subsequently acted upon using another clock input.

### MACROCELL ARCHITECTURE

The next level of detail to examine the iPLD610 is at the macrocell level. Figure 3 provides an overview of the macrocell architecture. There are several key features that make this device very useful to system designers by overcoming limitations of the standard PAL\*/GAL\*\*/22V10 capabilities. Every macrocell of

the iPLD610 has 8 sum of product terms for standard logic implementation. There is a separate product term for Asynchronous Register clear (Reset) for each register. In addition, there is a product term that can be programmed as either an asynchronous clock input to the register or as an output enable control. This asynchronous clocking capability for each macrocell provides yet another level of clocking flexibility to an already advanced architecture. The output enable control available on each macrocell allows any I/O pin to be tri-stated and provides for full control of pins used as both inputs and outputs. The feedback from each macrocell can be direct register feedback or pin feedback. Several macrocell options are detailed in the iPLD610 Data Sheet (Lit. No. 290455). Each macrocell can be configured to implement registered or combinational logic. Register types available include D, T, JK and RS. This provides designers register types not found in many of the common programmable logic devices. Register type selection is also a common way to reduce product term requirements (rather than just going to higher product term devices), as logic development tools can minimize equations for the best fit. Whether registered or combinational logic is implemented, there is also an inversion control within each macrocell. Again, this can help designers minimize logic and control output polarities by using DeMorgan's inversion rules on equations.

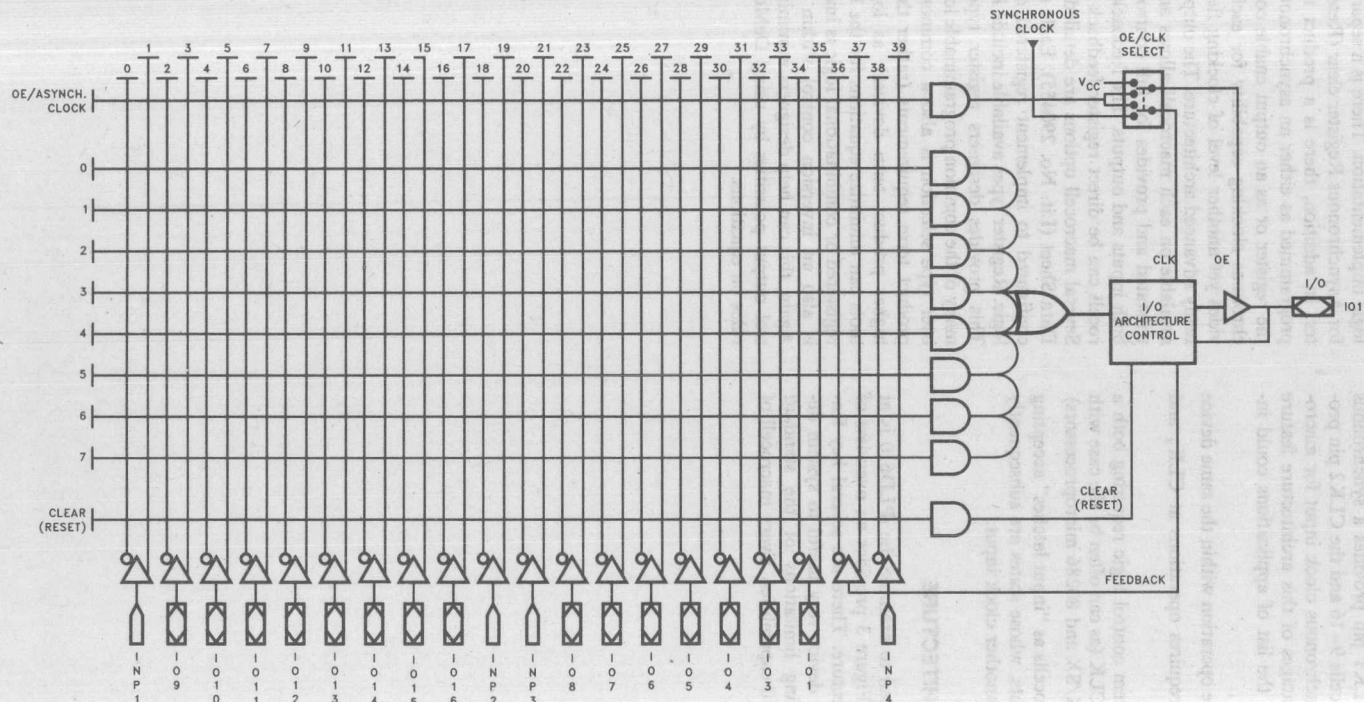
3



\*PAL® is a registered trademark of Advanced Micro Devices.

\*\*GAL® is a registered trademark of Lattice Semiconductor, Inc.





292074-4

Figure 3. IPLD610 Macrocell Architecture

### 3.0 SPECIFICATION ANALYSIS

When designing with any IC, a designer would like to know two things:

- 1) How is the part guaranteed to act (i.e., what are its specifications)
- 2) How will it act in its actual design environment

The first item above will be covered in this section. Using the data sheet A.C. and D.C. specifications a designer can perform a "paper analysis" of the circuit. The questions to be answered here are "How does the Intel iPLD610 compare with competitive devices" and "How can the A.C. and D.C. specs best be used?"

The second item is much harder to quantify due to the vast differences between designs. However, a great deal of bench data has been collected and presented in Section 4, which will help the designer make these assessments.

D.C. characteristics of the Intel iPLD610 are covered first. Next follows a comparison of these values with competitive devices. Following this is an overview of the A.C. specifications for the iPLD610 and, finally, a comparison of these specifications with other devices.

#### 3.1 D.C. Characteristics Analysis

##### TTL Compatibility

D.C. characteristics address the direct current components of operating any IC. The areas covered include input and output voltage and current levels, and  $I_{CC}$  supply current consumption.

The input voltage levels ( $V_{IH}$  and  $V_{IL}$ ) show that the iPLD610 can be driven by standard TTL or CMOS logic components, so the designer doesn't need to worry about mixing logic families—even though the iPLD610 is a CMOS PLD. The  $V_{OH}$  and  $V_{OL}$  show that the iPLD610 can drive TTL loads over the specified test conditions. CMOS loads are capacitive in nature and do not require much current (typically measured in  $\mu A$ ).

##### $I_{OL}$ Specifications

The  $V_{OL}$  test condition shows that each iPLD610 is capable of handling a 12 mA load while maintaining the output at or below 0.45V. Refer to Figure 17 (in Section 4) if another  $V_{OL}/V_{OH}$  value is required. That figure provides typical values for the output driver of the iPLD610.

Also in the D.C. specifications of the iPLD610 is a note specifying the total  $I_{OL}$  for each "bank" of 8 macrocells is 64 mA. This is the maximum recommended D.C. (steady state) current for this device. Even though the iPLD610 can sink 12 mA on each output, this 64 mA/bank limit should be observed to reduce electromigration effects. The duty cycle of each output should be incorporated when calculating the steady state device current. For example, each output of a 4-bit counter is active 50% of the time. The maximum current for each bank is 96 mA.

The input and output leakage current specifications are  $\pm 10 \mu A$  (per pin). This is an average value for CMOS devices. Typically, though, the total leakage for all pins (combined) will be within this  $\pm 10 \mu A$  range.

##### $I_{CC}$ Specifications

Probably the D.C. specification of the greatest concern to most designers is the  $I_{CC}$  value. This value tells the designer how much current will be required and how much heat (watts) will be generated by each device. Therefore, the  $I_{CC}$  specifications of a device will affect both the power supply requirements and the board reliability. To generate  $I_{CC}$  values, the device is tested with no load to find out how much power is consumed by the device itself.

Since CMOS devices consume most of their power during transitions,  $I_{CC}$  will greatly vary with the frequency of the clock or other inputs. For this reason not only are two values of  $I_{CC}$  specified in the D.C. characteristics (at 1 MHz), but an  $I_{CC}$  versus frequency graph is given at the back of the data sheet allowing more precise current/power calculations. This graph provides "typical"  $I_{CC}$  values (i.e.,  $V_{CC} = 5.0V$ ,  $temp = 25^\circ C$ ).

Another specification related to  $I_{CC}$  is the standby current specification,  $I_{SB}$ . This specification gives both typical and maximum values for power consumption when the iPLD610 is in standby mode.

To be in standby mode the device must be programmed with  $TURBO = OFF$  and no inputs can have changed state for 75 ns. Wakeup time adds an additional 25 ns to the propagation delay through the device as measured from the first input change (see Figure 4). The standby mode programming option provides designers of power-sensitive applications such as laptop PCs with the capability to greatly reduce the system's power budget.

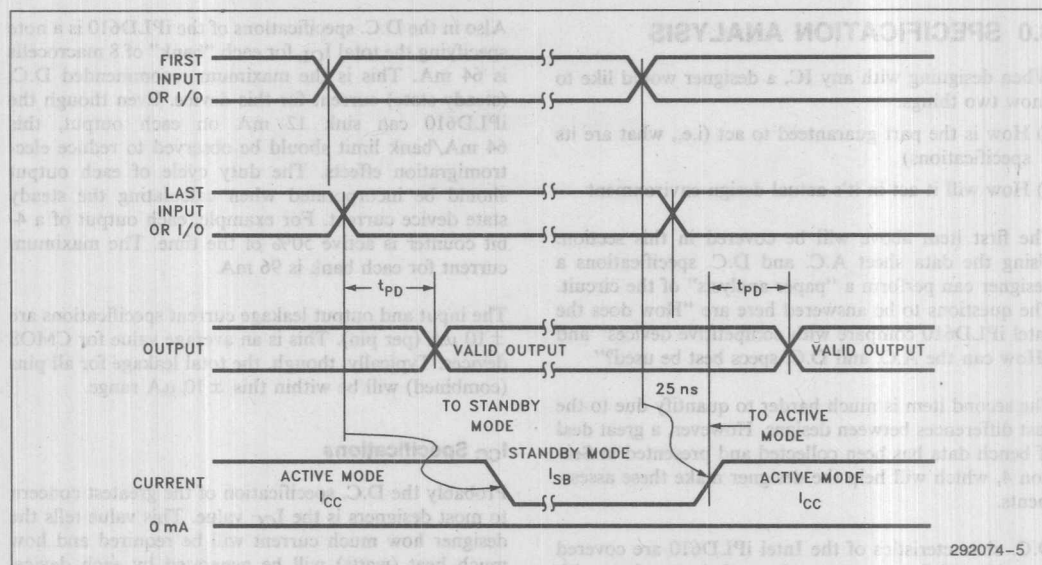


Figure 4. iPLD610 Standby and Active Mode Transitions

### 3.2 D.C. Specification Comparison

The voltage specifications for the iPLD610 and competitive devices do not differentiate where these devices can be used. For this reason Table 1, below, concentrates on the current specifications.

The conclusions from this table are clear. The Intel iPLD610 provides the lowest power consumption ( $I_{CC}$ ) while providing the highest output drive capability (12 mA). Also, its values for standby current are comparable to the other devices (although the AMD PALCE630 does not offer a standby mode).

Table 1. D.C. Specification Comparison

Spec	$I_{CC}$ (@ 1 MHz)		$I_{SB}$		Output Current	
	Typical	Max	Typ	Max	$I_{OL}$	$I_{OH}$
Part No.						
Intel iPLD610	3 mA	8 mA	20 $\mu$ A	100 $\mu$ A	12 mA	-4 mA
Intel 5C060	10 mA	15 mA	20 $\mu$ A	100 $\mu$ A	4 mA	-4 mA
Altera EP630	5 mA	10 mA	20 $\mu$ A	150 $\mu$ A	4 mA	-4 mA
Altera EP610	3 mA	10 mA	20 $\mu$ A	100 $\mu$ A	4 mA	-4 mA
TI EP610	3 mA	10 mA	20 $\mu$ A	100 $\mu$ A	4 mA	-4 mA
AMD PALCE630	Not Avail.	Not Avail.	Not Supported	Not Supported	8 mA	-4 mA

### 3.3 A.C. Specification Analysis

#### COMBINATORIAL

A.C. specifications provide the real “meat” of the data sheet by stating the guaranteed performance of the device. The first section of A.C. characteristics are the combinatorial mode A.C. characteristics. These state timings of non-registered logic paths in the iPLD610 including propagation delays ( $t_{PD1}$  and  $t_{PD2}$ ), output enable/disable times ( $t_{PZX}$  and  $t_{PXZ}$ ), and asynchronous reset time for any register ( $t_{CLR}$ ). These specifications are stating the times from any I/O or input pin until the selected function is performed. Propagation delays include AND/OR logic provided by up to eight product terms.

#### SYNCHRONOUS

The next table in the iPLD610 data sheet is the “Register Mode—Synchronous Clock A.C. Characteristics” which includes specifications related to using any macrocell in registered mode which is clocked with CLK1 or CLK2. The specifications of primary importance here revolve around the register setup time ( $t_{SU}$ ) and register clock to output time ( $t_{CO1}$ ). The setup time is the minimum time allowed between valid data appearing on an input and the active clock edge. The

clock to output time states the delay from the time data is clocked at the register until valid data shows up at the output pin. The reason these two specifications are of concern is that together they determine the worst case throughput performance in registered mode. One highlight of the iPLD610 is how closely they match standard “E-PAL” specifications. Section 3.4 includes a summary of iPLD610 versus PAL performance. The maximum counter frequency,  $F_{CNT1}$ , represents the maximum counter frequency with the iPLD610 using external feedback. External feedback means routing an output pin to another pin as an input. In this case the designer incurs both input and output buffer delays—but, this also simulates the activity of a multiple device counter or state machine. A related specification,  $F_{CNT2}$ , represents the maximum counter (state machine) frequency using internal device feedback.  $F_{CNT2}$  is higher than  $F_{CNT1}$  due to elimination of an input and output buffer from the feedback path. Figure 5 provides an overview of both  $F_{CNT}$  values.

The designer must also be aware that the  $F_{MAX}$  specification states the maximum frequency at which the registers in the iPLD610 can be clocked. This is due to the physical limitations on the period of the clock input (it could also be limited by register setup time). Therefore,  $F_{MAX} = 1/t_{CW}$  ( $= 1/\text{min clock width}$ ).

3

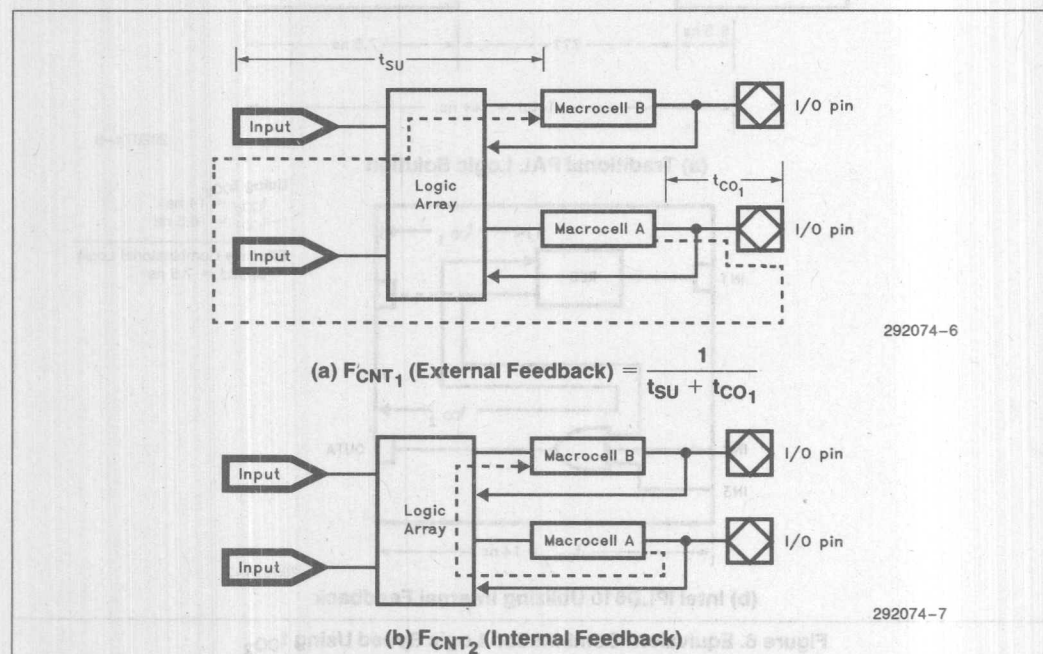


Figure 5.  $F_{CNT}$  Overview



Designs utilizing programmable logic can make full use of the iPLD610 by understanding the applications of  $FCNT_1$  (used for multiple device state machines),  $FCNT_2$  (used for smaller, single device state machines), and  $F_{MAX}$  (used for register pipelining applications).

Another A.C. specification of interest is  $t_{CO_2}$  which represents the time from CLK high to output valid fed through a combinatorial macrocell. Figure 6 depicts the difference between  $t_{CO_1}$  and  $t_{CO_2}$ . The  $t_{CO_2}$  specification shows the advantages of using internal feedback to gain speed when register output signals are used in combinatorial logic equations.

## ASYNCHRONOUS

For each of the synchronous clock A.C. characteristics there is an asynchronous clock A.C. characteristic specified in the data sheet. Each macrocell of the iPLD610 can be programmed to be clocked with an asynchronous clock (generated by a separate product term in each macrocell). The asynchronous clock values closely mirror the synchronous specifications in both definition and value, but, there are a few notable differences.

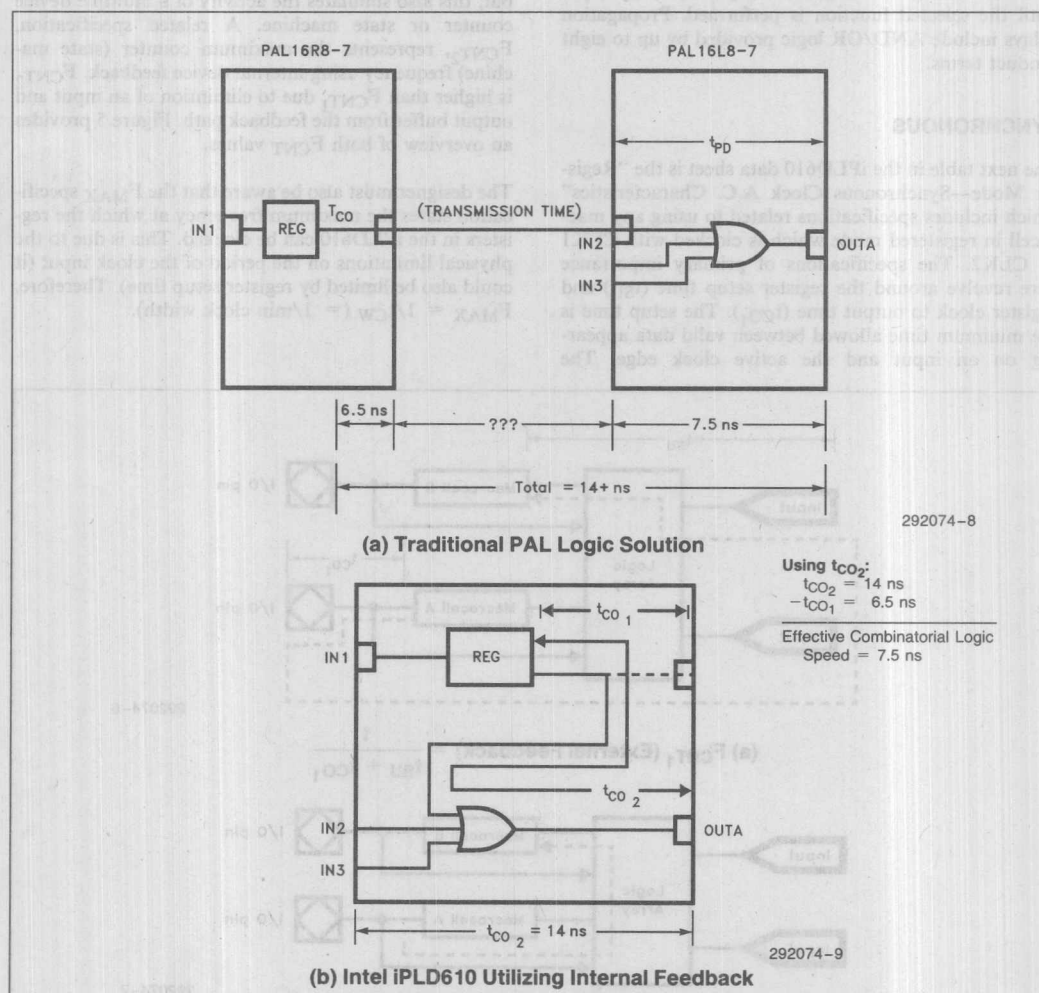


Figure 6. Equivalent Combination Logic Speed Using  $t_{CO_2}$

The synchronous register setup and hold time specifications ( $t_{SU}$  and  $t_H$ ) are 7 ns and 0 ns, respectively. The asynchronous register specifications show a shifting of specifications due to differences in internal paths. The asynchronous register setup and hold time specifications ( $t_{ASU}$  and  $t_{AH}$ ) are 2 ns and 3 ns, respectively. A similar shifting is seen if register setup and clock-to-output specifications are compared for synchronous and asynchronous operation as detailed in Figure 7.

Understanding of these differences between synchronous and asynchronous specifications can be useful to a designer. If a system design requires a short setup time

(in the 2 ns–5 ns range) one or more of the macrocells of the iPLD610 could be used in the asynchronous clocking mode. Either a control input or the system's synchronous clock could be routed to an input or I/O pin to establish the necessary "asynchronous" clock. With this technique, setup times as low as 3 ns can be met by the iPLD610, although there is the trade off of a longer register clock-to-output delay. Low setup times are often required to accommodate output valid delay specifications of many microprocessors and system peripherals.

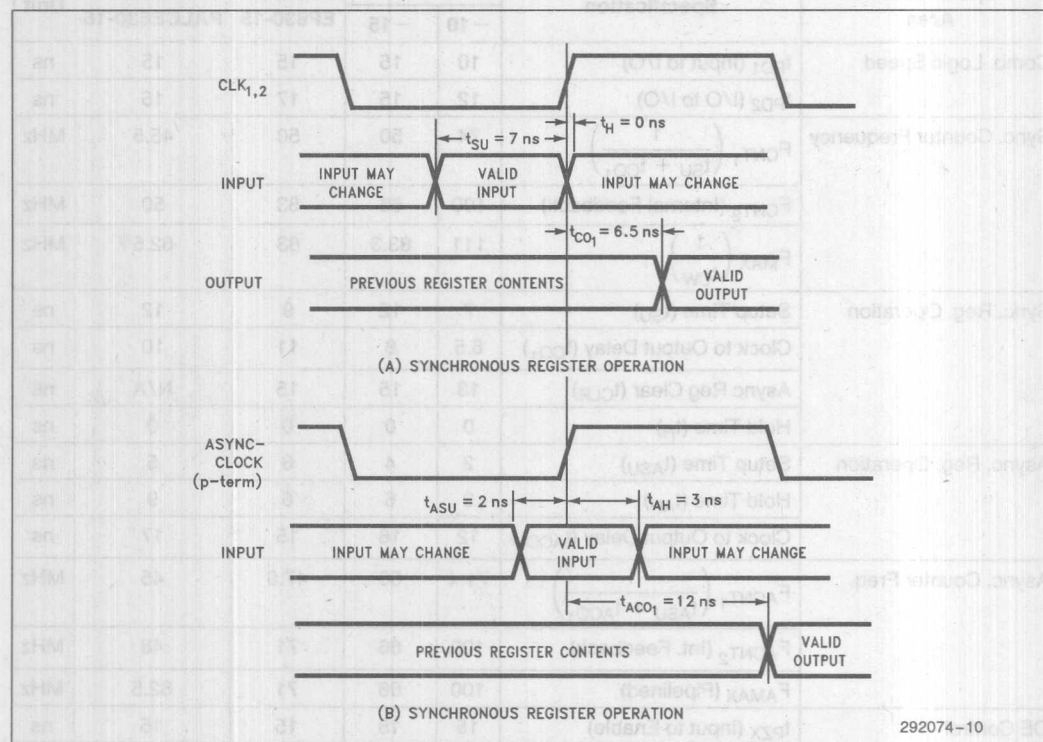


Figure 7. iPLD610 Register Timings

### 3.4 A.C. Specifications Comparison

The comparison made here involves other devices with the same architecture as the Intel iPLD610. This includes devices from Altera, TI and AMD. It should be noted that Tables 2 and 3 (below) do not compare all A.C. specifications, rather only those best reflecting the performance of these devices. Even though the architectures of these devices (i.e., pinout, product term,

macrocell configurations, etc.) are all exactly the same there are some differences in the test conditions used to generate A.C. specification. These differences are noted.

The first comparison involves the higher speed parts—those in the 10 ns–15 ns range for  $t_{PD}$ . The second comparison will involve slower parts—those with a  $t_{PD}$  of 25 ns. The final comparison shows a summary of iPLD610 performance versus standard PAL devices.

Table 2. A.C. Specification Comparison—High Performance

Specification Area	Specification	iPLD610		Altera EP630-15	AMD PALCE630-15	Unit
		– 10	– 15			
Comb. Logic Speed	$t_{PD1}$ (Input to I/O)	10	15	15	15	ns
	$t_{PD2}$ (I/O to I/O)	12	15	17	15	ns
Sync. Counter Frequency	$F_{CNT1} \left( \frac{1}{t_{SU} + t_{CO1}} \right)$	74	50	50	45.5	MHz
	$F_{CNT2}$ (Internal Feedback)	100	66	83	50	MHz
	$F_{MAX} \left( \frac{1}{t_{CW}} \right)$	111	83.3	83	62.5	MHz
Sync. Reg. Operation	Setup Time ( $t_{SU}$ )	7	12	9	12	ns
	Clock to Output Delay ( $t_{CO1}$ )	6.5	8	11	10	ns
	Async Reg Clear ( $t_{CLR}$ )	13	15	15	N/A	ns
	Hold Time ( $t_H$ )	0	0	0	0	ns
Async. Reg. Operation	Setup Time ( $t_{ASU}$ )	2	4	6	5	ns
	Hold Time ( $t_{AH}$ )	3	6	6	9	ns
	Clock to Output Delay ( $t_{ACO1}$ )	12	16	15	17	ns
Async. Counter Freq.	$F_{ACNT1} \left( \frac{1}{t_{ASU} + t_{ACO1}} \right)$	71.4	50	47.6	45	MHz
	$F_{ACNT2}$ (Int. Feedback)	100	66	71	48	MHz
	$F_{AMAX}$ (Pipelined)	100	66	71	62.5	MHz
OE Control	$t_{PZX}$ (Input to Enable)	15	18	15	15	ns
	$t_{PXZ}$ (Input to Disable)	13	18	15	15	ns

**Table 3. A.C. Specification Comparison—Mid Range Device Propagation Delay = 25 ns**

Specification Area	Specification	iPLD610-25	Altera/TI EP610-25	AMD PALCE630-25	Unit
Comb. Logic Speed	t <sub>PD1</sub> (Input to I/O)	25	25	25	ns
	t <sub>PD2</sub> (I/O to I/O)	25	27	25	ns
	t <sub>CLR</sub> (Async Reg Clear)	25	27	N/A	ns
Sync. Register Performance	$F_{CNT1} \left( \frac{1}{t_{SU} + t_{CO1}} \right)$	40	27.8	37	MHz
	F <sub>CNT2</sub> (Int Feedback)	40	40	40	MHz
	$F_{MAX} \left( \text{Pipelined} = \frac{1}{t_{CW}} \right)$	50	47.6	50	MHz
Sync Register Operation	t <sub>SU</sub> (Setup Time)	15	21	15	ns
	t <sub>H</sub> (Hold Time)	0	0	0	ns
	t <sub>CO1</sub> (Clock to Output Delay)	10	15	12	ns
Async Register Performance	F <sub>ACNT1</sub> (Ext. Feedback)	33.3	28.5	28.6	MHz
	F <sub>ACNT2</sub> (Int. Feedback)	40	40	29	MHz
	F <sub>AMAX</sub> (Pipelined)	50	47.6	41.6	MHz
Async Register Operation	t <sub>ASU</sub> (Setup Time)	5	8	8	ns
	t <sub>AH</sub> (Hold Time)	8	12	12	ns
	t <sub>ACO1</sub> (Clock to Output Delay)	25	27	27	ns
OE Control	t <sub>PZX</sub> (Input to Enable)	25	25	25	ns
	t <sub>PXZ</sub> (Input to Disable)	25	25	25	ns



At the "slower" end of the product line ( $t_{PD} = 25$  ns), the iPLD610 provides an upgrade to existing 25 ns  $t_{PD}$  devices—especially in the area of synchronous register operations. From there a designer can migrate a design to the iPLD610-15 and finally to the state-of-the-art in architecture/performance, the Intel iPLD610-10.

### PAL Comparison

A common application for higher integration devices such as the iPLD610  $\mu$ PLD is upgrading existing PAL designs. Table 4 summarizes the performance specifications of the iPLD610 and standard PAL devices. The primary register-related specifications of the iPLD610,  $t_{SU}$  and  $t_{CO1}$ , are equal to E-PAL performance levels. This results in  $F_{CNT1}$  specifications between the two devices which are very close. Also of note is that  $F_{MAX}$  of the iPLD610 is 111 MHz, compared to 100 MHz for the E-PAL. The major differences between the iPLD610 and the PAL devices are in the  $t_{PD}$  and  $I_{CC}$  values. The iPLD610 cannot meet the faster combinational logic speeds of PAL devices, however, PALs require about 2 times more current than the iPLD610. This  $I_{CC}$  difference is important in many design areas including power supply sizing, cooling requirements, and overall system reliability.

**Table 4. iPLD610 vs PAL  
Performance Summary**

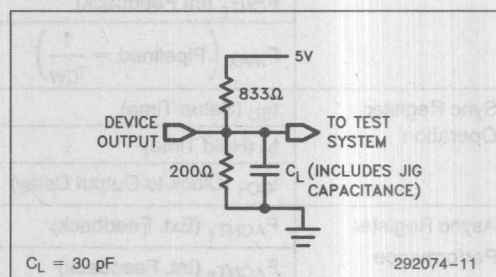
Specification	iPLD610	"E-PAL" 20xx-7	"D-PAL" 20xx-10	Units
$t_{SU}$	7	7	10	ns
$t_{CO1}$	6.5	6.5	8	ns
$F_{CNT1}$	74	74	55.5	MHz
$F_{MAX}$	111	100	71.4	MHz
$t_{PD}$	10	7.5	10	ns
$I_{CC}(\max)$	105	210	210	mA

### 4.0 ADVANCED DESIGN ISSUES

As system designs climb to higher speeds and time to market becomes more critical, designers require more detailed information than is available in the data sheet. The main emphasis here is to provide designers with characterization data of key aspects of the iPLD610. This data will help avoid unforeseen problems in the design, prototype, and production phases.

Data was measured with the output load specified in the iPLD610 data sheet (shown also in Figure 8), unless otherwise specified. The topics covered in this section are:

- Output Slew Rates
- Combinational Logic Concerns ( $t_{PD}$  Characteristics)
- Synchronous Register Operation Characteristics
- Asynchronous Register Operation Characteristics
- Output Current Characteristics
- Design Considerations



**Figure 8. A.C. Testing Load Circuit**

### Output Slew Rates

Several key advantages of using Intel  $\mu$ PLDs are derivatives of the output buffer design. One of these is slow output slew rates which minimizes system noise. Signal ringing (noise) can result if output rise times are shorter than twice the signal transmission time. Thus, the faster the edge rate of any device output the shorter the board trace that is allowable before transmission line effects must be considered. This generally involves series or parallel termination and more consideration of device location and signal routing. The iPLD610 seeks to limit these design problems by providing lower output slew (edge) rates.

Figures 9 and 10 provide a sample of the iPLD610 output slew rate characterization. Table 5 shows there is very little variation over temperature, although the values do decrease as temperature increases. Table 6 shows there is very little variation over the number of outputs switching simultaneously, although it does decrease as the number of outputs increases. The results show output slew rates comparable to those of bipolar PALs and much lower than CMOS GALs (which are in the 3–5 V/ns range). The edge rates provided by the iPLD610 provide designers with flexibility without constantly worrying about transmission line effects.

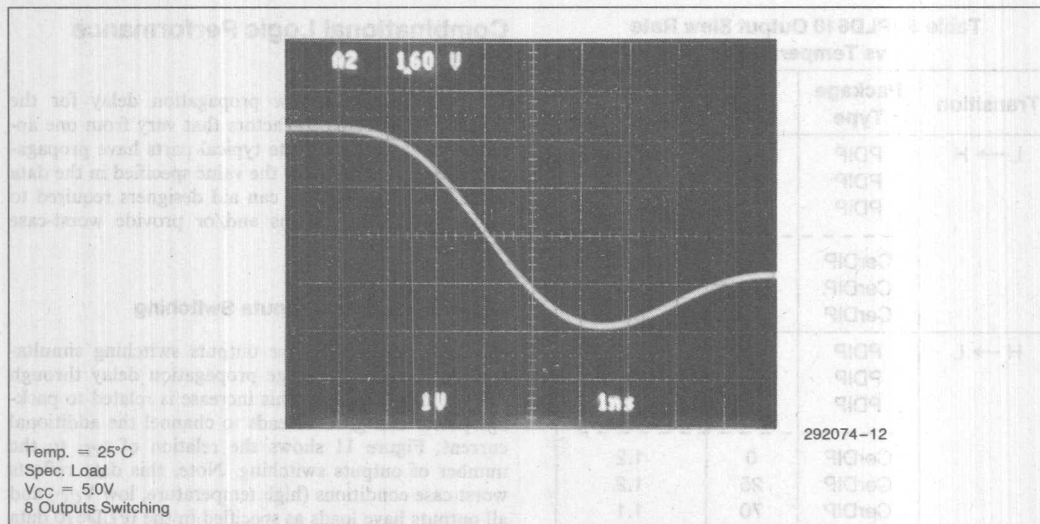


Figure 9. iPLD610 Output Slew—H → L Transition

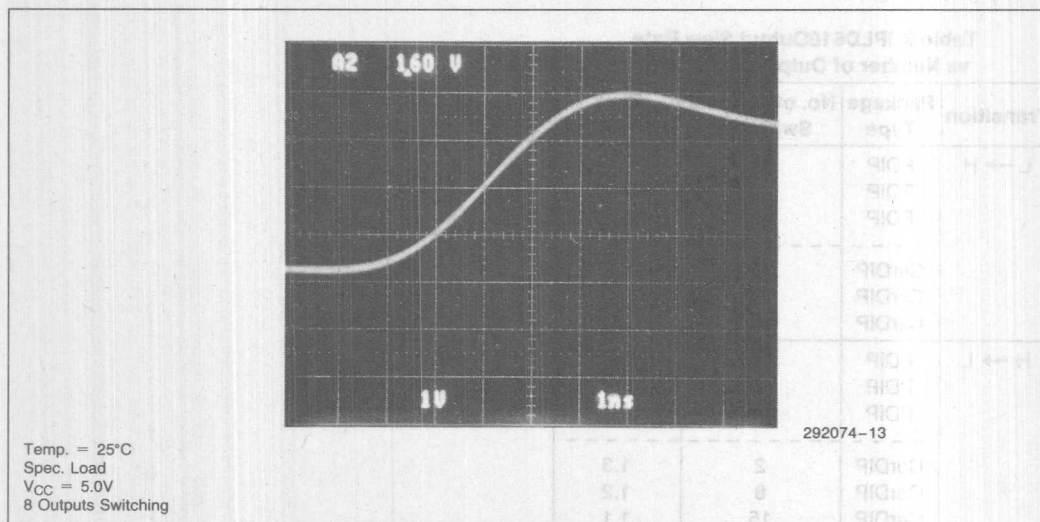


Figure 10. iPLD610 Output Slew—L → H Transition

**Table 5. iPLD610 Output Slew Rate  
vs Temperature**

Transition	Package Type	Temp (°C)	Slew Rate (V/ns)
L → H	PDIP	0	1.1
		25	1.1
		70	1.0
	CerDIP	0	1.0
		25	1.0
		70	0.9
H → L	PDIP	0	1.2
		25	1.2
		70	1.1
	CerDIP	0	1.2
		25	1.2
		70	1.1

$V_{CC} = 5.0V$

No outputs switching = 8

**Table 6. iPLD610 Output Slew Rate  
vs Number of Outputs Switching**

Transition	Package Type	No. of Outputs Switching	Slew Rate (V/ns)
L → H	PDIP	2	1.1
		8	1.1
		15	1.0
	CerDIP	2	1.1
		8	1.0
		15	0.9
H → L	PDIP	2	1.3
		8	1.2
		15	1.2
	CerDIP	2	1.3
		8	1.2
		15	1.1

$V_{CC} = 5.0V$

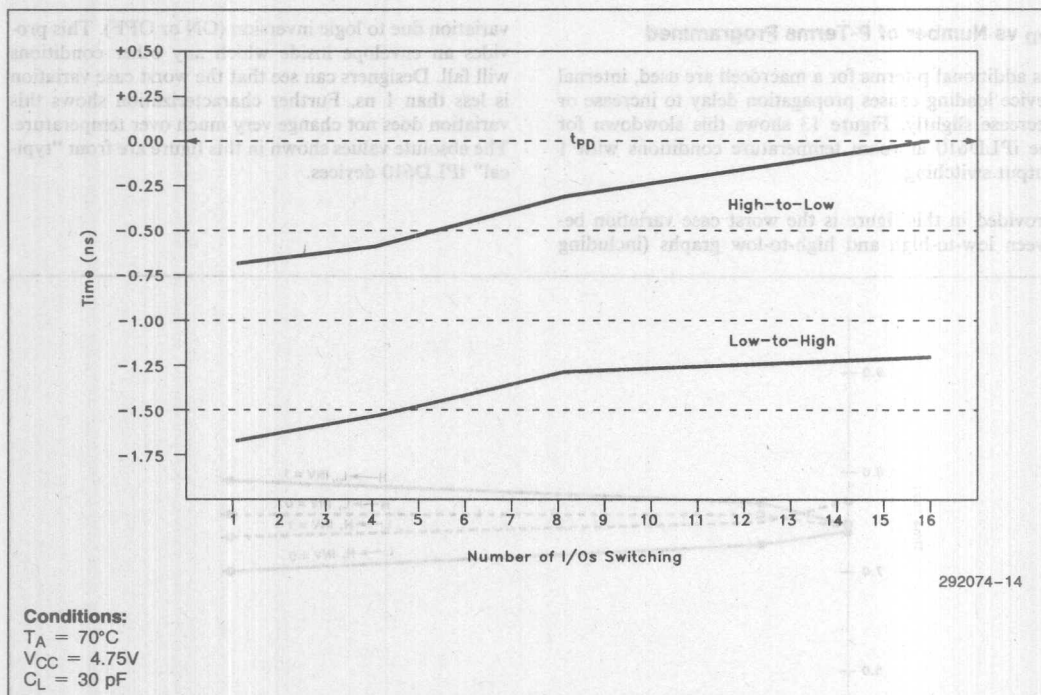
Temp. = 25°C

## Combinational Logic Performance ( $t_{PD}$ )

This section shows how propagation delay for the iPLD610 is affected by factors that vary from one application to another. Note typical parts have propagation delays 1–2 ns below the value specified in the data sheet. Data such as this can aid designers required to “fine tune” their designs and/or provide worst-case timing analyses.

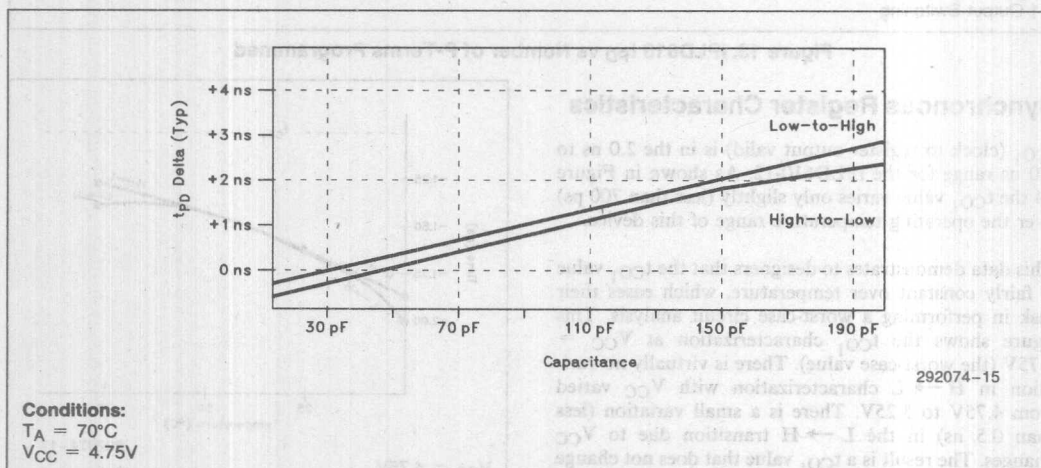
### $t_{PD}$ vs Number of Outputs Switching

As the number of device outputs switching simultaneously increases, average propagation delay through devices also increases. This increase is related to package power and ground leads to channel the additional current. Figure 11 shows the relation of  $t_{PD}$  to the number of outputs switching. Note, this data reflects worst case conditions (high temperature, low  $V_{CC}$ ) and all outputs have loads as specified in the iPLD610 data sheet.

Figure 11. iPLD610  $t_{PD}$  vs Number of Outputs Switching $t_{PD}$  vs  $C_L$ 

Knowledge of how devices behave as capacitive loading is increased is an important consideration when designing high-speed systems. Figure 12 shows derating from

specified values for a typical iPLD610 at high temperature, low  $V_{CC}$  conditions for both low-to-high and high-to-low transitions as capacitance increases.

Figure 12. iPLD610  $t_{PD}$  vs  $C_L$



### $t_{PD}$ vs Number of P-Terms Programmed

As additional p-terms for a macrocell are used, internal device loading causes propagation delay to increase or decrease slightly. Figure 13 shows this slowdown for the iPLD610 at room temperature conditions with 1 output switching.

Provided in this figure is the worst case variation between low-to-high and high-to-low graphs (including

variation due to logic inversion (ON or OFF). This provides an envelope inside which any other conditions will fall. Designers can see that the worst case variation is less than 1 ns. Further characterization shows this variation does not change very much over temperature. The absolute values shown in this figure are from "typical" iPLD610 devices.

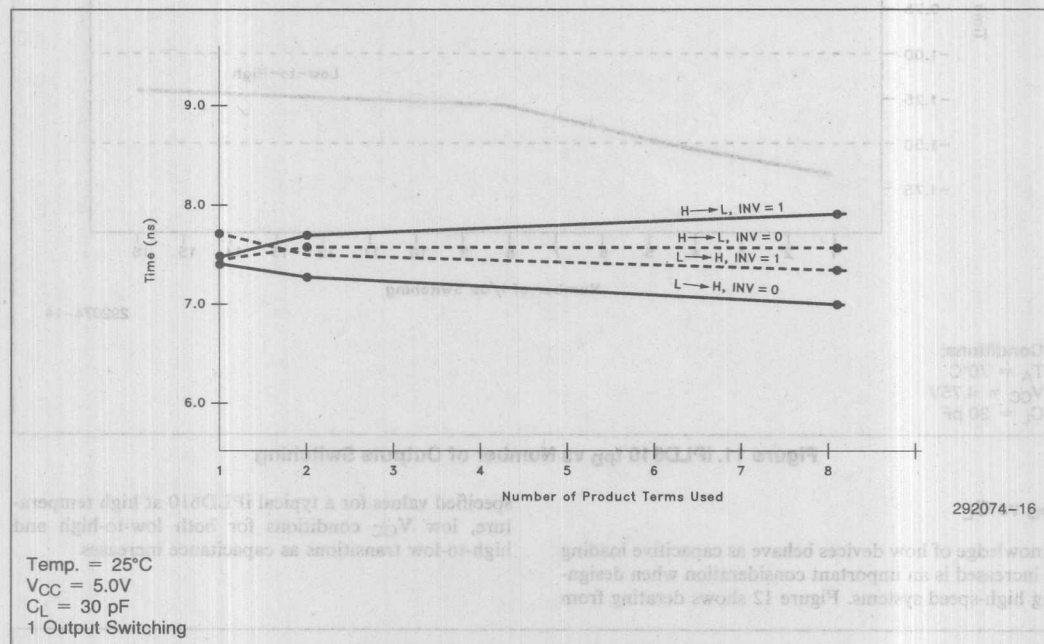


Figure 13. iPLD610  $t_{PD}$  vs Number of P-Terms Programmed

### Synchronous Register Characteristics

$t_{CO1}$  (clock to register output valid) is in the 2.0 ns to 7.0 ns range for the iPLD610-12. As shown in Figure 14 the  $t_{CO1}$  value varies only slightly (less than 700 ps) over the operating temperature range of this device.

This data demonstrates to designers that the  $t_{CO1}$  value is fairly constant over temperature, which eases their task in performing a worst-case circuit analysis. This figure shows the  $t_{CO1}$  characterization at  $V_{CC} = 4.75V$  (the worst-case value). There is virtually no variation in  $H \rightarrow L$  characterization with  $V_{CC}$  varied from 4.75V to 5.25V. There is a small variation (less than 0.5 ns) in the  $L \rightarrow H$  transition due to  $V_{CC}$  changes. The result is a  $t_{CO1}$  value that does not change much over temperature and  $V_{CC}$ .

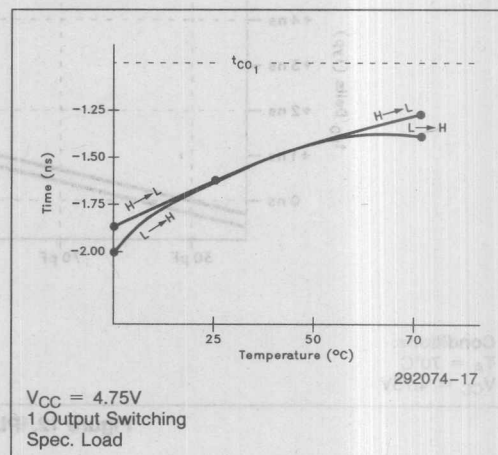


Figure 14.  $t_{CO1}$  vs Temperature

### Register-to-Register Skew

When registers within the same programmable logic device are clocked (using the synchronous clock input), ideally all outputs would change state simultaneously. This does not reflect the reality of differences in internal clock routing and ground path differences. Thus, there is some skew between outputs. As long as this skew remains small there is no impact on system design. Large skews can cause a need for additional synchronization logic and re-evaluation of system timing constraints.

Due to its high-speed double metal process, the output skew on the iPLD610 is very tight. Typical skew between fastest and slowest register within one "bank" (i.e., clocked by the same clock pin) is shown in Table 7.

**Table 7. iPLD610 Register-to-Register Skew Characterization**

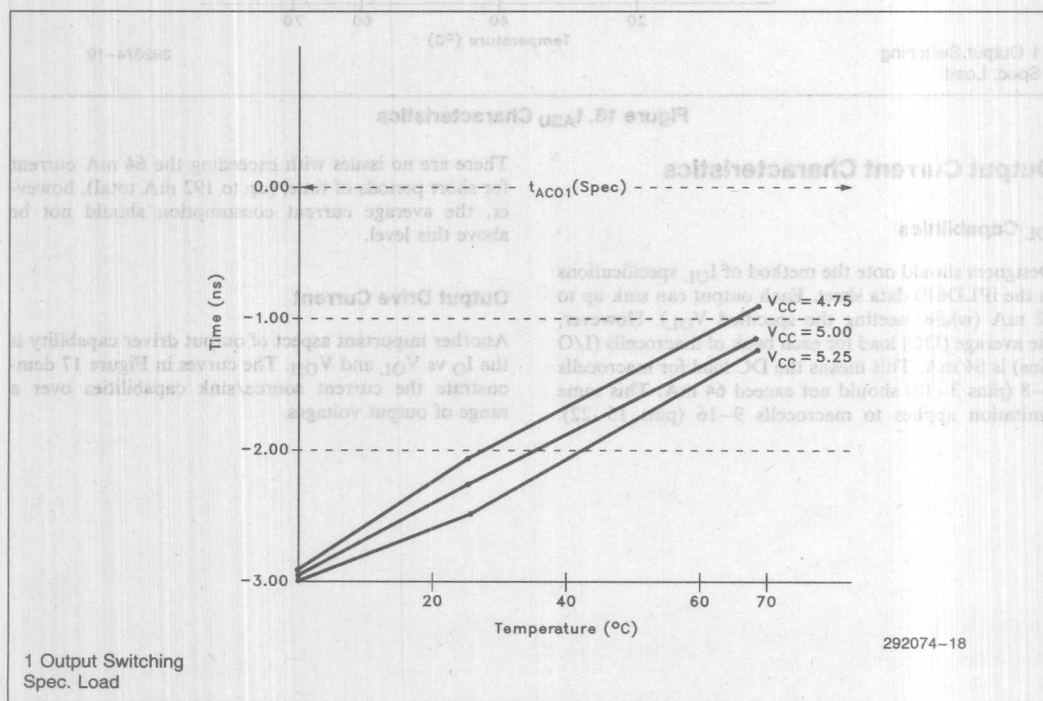
0°C		70°C	
High to Low (ps)	Low to High (ps)	High to Low (ps)	Low to High (ps)
120	310	120	340

### ASYNCHRONOUS REGISTER OPERATION CHARACTERISTICS

In addition to combinational logic and synchronous registered logic, the iPLD610 can implement asynchronously-clocked registered logic. This means a product (AND) term can be used to control the register in any macrocell. Each macrocell has a separate product term which can be used for this purpose. Also, as discussed in Section 3 there are a separate set of A.C. specifications for asynchronous register operation. Two of the key specifications are the asynchronous clock-to-output delay ( $t_{ACO1}$ ) and the asynchronous register setup time ( $t_{ASU}$ ).

#### $t_{ACO1}$ Characteristics

Knowing how this specification varies over supply voltage ( $V_{CC}$ ) and temperature may be useful to a designer concerned with detailed system timing analysis. Figure 15 shows this  $t_{ACO1}$  characterization (H  $\rightarrow$  L transition only) and how small the changes are over both of these variables. This figure shows  $t_{CO1}$  will increase if  $V_{CC}$  is decreased or if system temperature increases, although temperature has a much greater impact on its value. The total variation over both of these parameters is fairly small (less than 2 ns) showing the solid design of the asynchronous circuitry of the iPLD610.



**Figure 15. iPLD610  $t_{ACO1}$  Characteristics (H  $\rightarrow$  L)**

### $t_{ASU}$ Characteristics

Another important specification related to asynchronous register operation is  $t_{ASU}$ , the register setup time. A designer performing circuit analysis may need to know how this may vary over temperature and supply

voltage. Figure 16 presents the characterization of this specification. Of note are the very small changes over both temperature (less than 50 ps over 0°C–70°C range) and supply voltage (less than 350 ps variation over a 4.75V–5.25V range). Both of these show the exceptional stability of this circuitry.

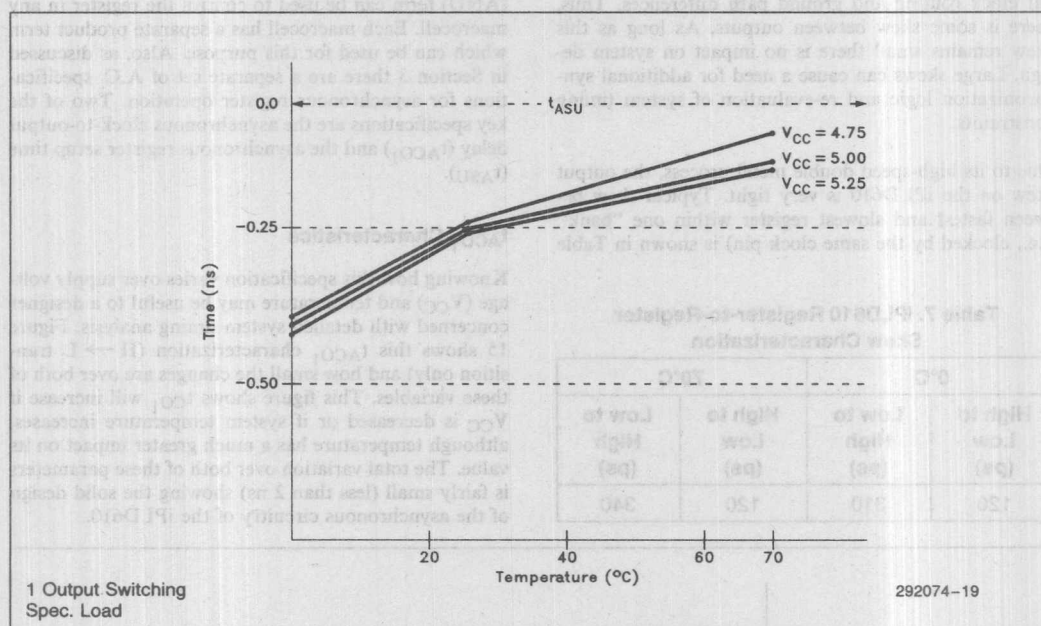


Figure 16.  $t_{ASU}$  Characteristics

### Output Current Characteristics

#### $I_{OL}$ Capabilities

Designers should note the method of  $I_{OL}$  specifications in the iPLD610 data sheet. Each output can sink up to 12 mA (while meeting the specified  $V_{OL}$ ). However, the average (DC) load for each bank of macrocells (I/O pins) is 64 mA. This means the DC load for macrocells 1–8 (pins 3–10) should not exceed 64 mA. This same limitation applies to macrocells 9–16 (pins 15–22).

There are no issues with exceeding the 64 mA current for short periods of time, (up to 192 mA total), however, the average current consumption should not be above this level.

#### Output Drive Current

Another important aspect of output driver capability is the  $I_O$  vs  $V_{OL}$  and  $V_{OH}$ . The curves in Figure 17 demonstrate the current source/sink capabilities over a range of output voltages.

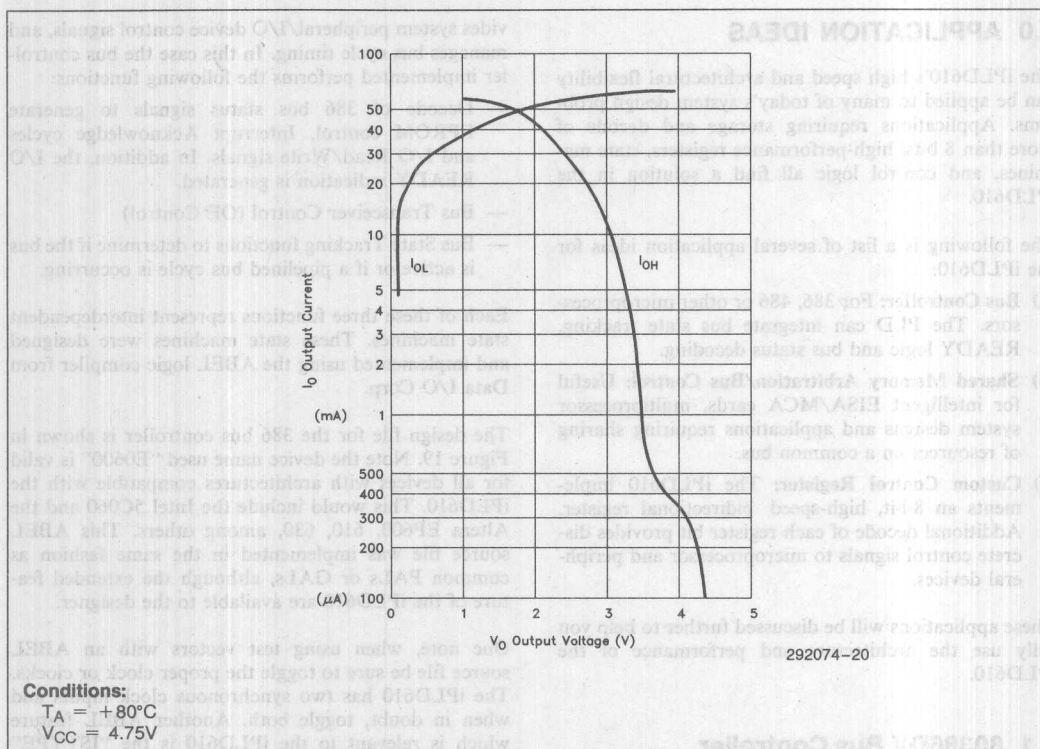


Figure 17. iPLD610 Output Drive Current

### Other Design Considerations

High-performance CMOS devices such as the iPLD610 are capable of driving large loads at fast edge rates. This capability means that noise control must be an important consideration during system design. Multi-layer PC boards utilizing ground and power planes provide low resistance and low inductance connections between the power sources and devices.

System noise control also requires good decoupling capacitance. Boards with power and ground planes but no decoupling capacitors can still have noise problems. High speed transients of devices may demand up to 500 mA of current, which can result in a volt or more of switching noise on the local supply lines. Decoupling capacitors can help prevent this performance degradation by providing a local power source during output transitions. With the addition of decoupling capacitors, it is possible to reduce the local supply noise to 200 mV or less.

Capacitor selection is important for this application since the frequencies involved in high-speed systems can exceed 100 MHz. High-frequency capacitors are called for. The capacitors should provide low series inductance; leadless chip caps are the best choice, with

short leaded capacitors as a more available second choice. The equivalent circuit for a capacitor is a series resonant circuit. If the inductive element in the capacitor is too high, the capacitor will appear inductive at high frequencies.

Assuming that everything possible has been done to manage noise on the supply lines outside the device, internal noise can still be a problem. The internal noise generated during switching transients is caused by output buffer design, package design, and output loading. Some suggestions for reducing noise are as follows:

- Select a low-inductance package such as PLCC.
- Reduce the output loading.
- Reduce the number of simultaneously switching outputs.
- Limit the voltage swing to 0V–3V by correctly terminating outputs with resistors to ground.

Designing high-speed P.C. boards requires closer attention to design issues that are not as important for slower systems. These elements include:

- Termination of transmission lines.
- Clock signal routing.
- Power distribution and heat dissipation.



## 5.0 APPLICATION IDEAS

The iPLD610's high speed and architectural flexibility can be applied to many of today's system design problems. Applications requiring storage and decode of more than 8 bits, high-performance registers, state machines, and control logic all find a solution in the iPLD610.

The following is a list of several application ideas for the iPLD610:

- (a) **Bus Controller:** For 386, 486 or other microprocessors. The PLD can integrate bus state tracking, READY logic and bus status decoding.
- (b) **Shared Memory Arbitration/Bus Control:** Useful for intelligent EISA/MCA cards, multiprocessor system designs and applications requiring sharing of resources on a common bus.
- (c) **Custom Control Register:** The iPLD610 implements an 8-bit, high-speed, bidirectional register. Additional decode of each register bit provides discrete control signals to microprocessor and peripheral devices.

These applications will be discussed further to help you fully use the architecture and performance of the iPLD610.

### 5.1 80386DX Bus Controller

In every 80386DX microprocessor system, bus control logic must be implemented to provide an interface to system peripherals, I/O devices, and system memory. The bus controller requirements can vary in complexity depending on system performance, memory hierarchy, and other factors. Figure 18 shows an example of an 80386 subsystem with an iPLD610 bus controller. The bus controller decodes processor status signals, pro-

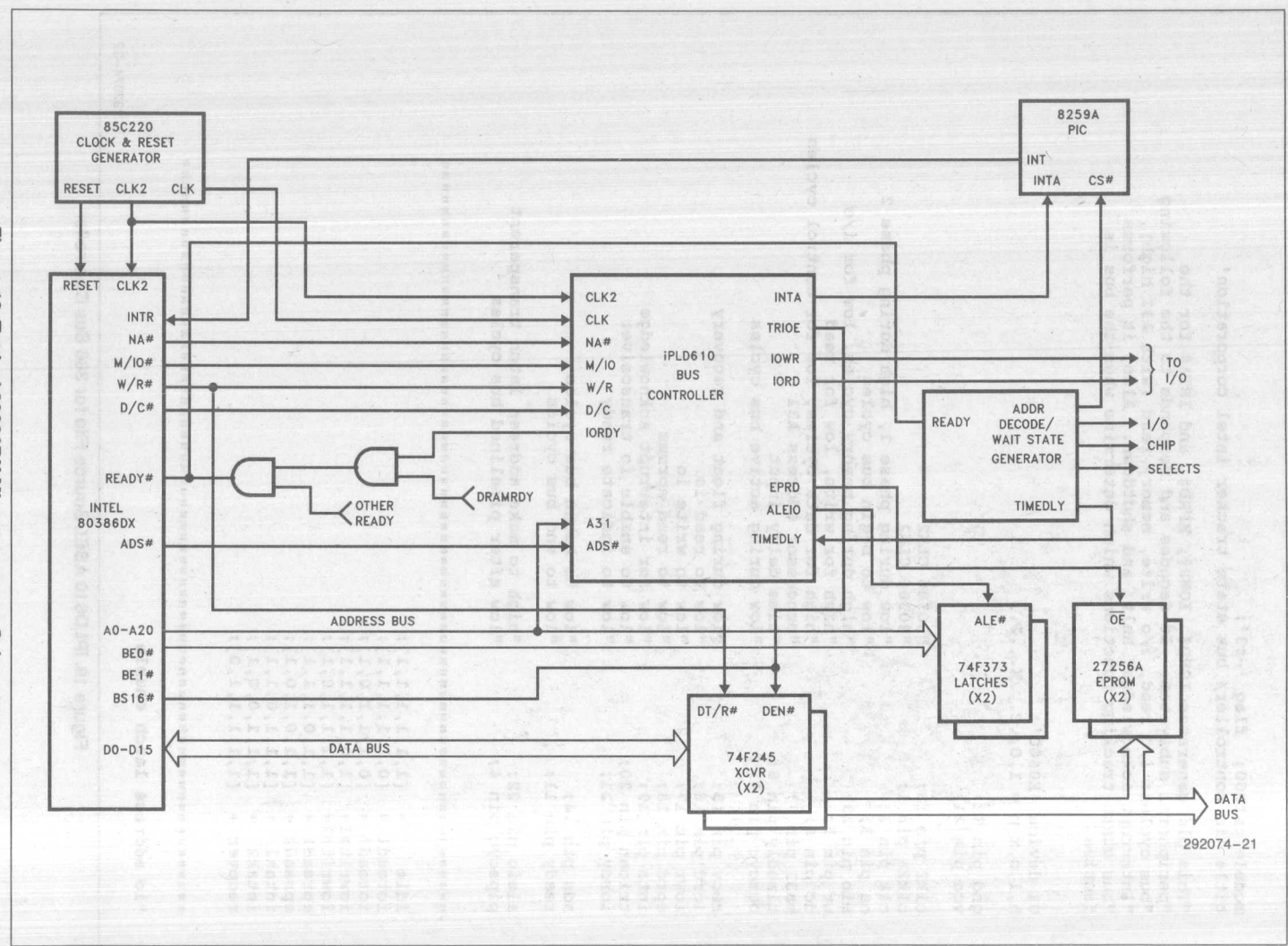
vides system peripheral/I/O device control signals, and manages bus cycle timing. In this case the bus controller implemented performs the following functions:

- Decode of 386 bus status signals to generate EPROM control, Interrupt Acknowledge cycles and I/O Read/Write signals. In addition, the I/O READY indication is generated.
- Bus Transceiver Control (OE Control)
- Bus State Tracking functions to determine if the bus is active or if a pipelined bus cycle is occurring.

Each of these three functions represent interdependent state machines. These state machines were designed and implemented using the ABEL logic compiler from Data I/O Corp.

The design file for the 386 bus controller is shown in Figure 19. Note the device name used "E0600" is valid for all devices with architectures compatible with the iPLD610. This would include the Intel 5C060 and the Altera EP600, 610, 630, among others. This ABEL source file was implemented in the same fashion as common PALs or GALs, although the extended feature of the iPLD610 are available to the designer.

One note, when using test vectors with an ABEL source file be sure to toggle the proper clock or clocks. The iPLD610 has two synchronous clock inputs and when in doubt, toggle both. Another ABEL feature which is relevant to the iPLD610 is the "ISTYPE" statement. This can be used by the designer if specific register types, feedback or invert options are required for any iPLD610 implementation. Upon successful compilation of this source file by ABEL, a .DOC file (see Figure 20) was generated (along with the required JEDEC file) which shows the reduced equations and device pin-out. This bus state tracker will work for 80386DX designs up to 33 MHz (CLK2 = 66 MHz) due to the iPLD610's state machine frequency specification (FCNT<sub>1</sub>) of 66 MHz.



292074-21

Figure 18. Typical 80386DX Microprocessor Subsystem

3-41

```

module pst060; flag '-r3';
title 'io controller/ bus state tracker intel corporation'

"This pld generates IORD#, IOWR#, EPRD#, and INTA# for the
"peripheral subsystem. It decodes and responds to the following
"bus cycles: i/o read, i/o write, memory read (with A31 high),
"interrupt acknowledge, halt, and shutdown. Also, it performs
"bus state tracking functions which determine when the bus is
"active.

U1 device 'E0600';
h,1,c,x,p = 1,0,.C.,.X.,.P.;

gnd pin 12;
vcc pin 24;

clk2 pin 13;          "80386 CLK2
clk2a pin 1;          "80386 CLK2
clk pin 2;            "low during phase 1, high during phase 2
na pin 3;             "low to begin bus cycles
mio pin 23;           "high during memory cycles, low for i/o
wr pin 5;             "high for write, low for read
ic pin 6;             "high for data cycles, low for control cycles
pa31 pin 7;           "processor address A31
timedly pin 8;        "time delay input
buscyc pin 9;         "low during active bus cycles

recv pin 15;          "low during float and recovery
iord pin 16;          "low to read io
iowr pin 17;          "low to write io
eprd pin 18;          "low to read eproms
inta pin 19;          "low for interrupt acknowledge
trioen pin 20;        "low to enable io transceiver
iordy pin 21;         "low to indicate ready

ads pin 14;           "low to begin bus cycles
ready pin 11;         "low to end bus cycles

aleio pin 22;         "high to make address latch transparent
pipecyc pin 4;        "low after pipelined bus cycles

--
idle   = [1,1,1,1,1,1];
ioread1 = [0,1,1,1,1,1];
ioread2 = [0,1,1,1,0,1];
iowrite1 = [1,0,1,1,1,1];
iowrite2 = [1,1,1,1,0,1];
epread1 = [1,1,0,1,1,1];
epread2 = [1,1,0,1,0,1];
intak1 = [1,1,1,0,1,1];
intak2 = [1,1,1,0,0,1];
recover = [1,1,1,1,1,0];

--

"io address latch enable

```

292074-32

Figure 19. iPLD610 ABEL Source File for 386 Bus Controller

```

equations !aleio := (!iord & clk) #
                  (!iowr & clk) #
                  (!inta & clk) #
                  (!aleio & !clk);

"io transceiver enable

state_diagram [trioen];
state 1: "idle
    if (na & !buscyc & !mio & !pa31 & recv & clk) then 0
    else if (na & !buscyc & mio & pa31 & recv & clk) then 0
    else 1;

state 0: "enable transceiver between processor and peripherals
    if (!iordy & clk) then 1
    else if (buscyc & clk) then 1
    else 0;

--
"io state machine
state_diagram [iord, iowr, eprd, inta, iordy, recv];
state idle:
    case na & !buscyc & pa31 & !wr & clk: epread1;
        na & !buscyc & !pa31 & !mio & dc & wr & clk: iowritel;
        na & !buscyc & !pa31 & !mio & dc & !wr & clk: ioread1;
        na & !buscyc & !pa31 & !mio & !dc & !wr & clk: intak1;
        na & !buscyc & mio & !dc & wr & clk: iowrite2; "halt
    endcase;

state epread1: if (!timedly & clk) then epread2 else epread1;
state epread2: if (clk) then idle else epread2;
state iowritel: if (!timedly & clk) then iowrite2 else iowritel;

state iowrite2: if (!mio & clk) then recover
    else if (mio & clk) then idle
    else iowrite2;
state ioread1: if (!timedly & clk) then ioread2 else ioread1;
state ioread2: if (clk) then recover else ioread2;
state intak1: if (!timedly & clk) then intak2 else intak1;
state intak2: if (clk) then recover else intak2;
state recover: if (!timedly & clk) then idle else recover;

--
"bus cycle tracking
state_diagram [buscyc, pipecyc]
state [1,1]: "idle
    if (!ads & clk) then [0,1]
    else [1,1];

state [0,1]: "active
    if (!ready & ads & clk) then [1,1]
    else if (!ready & !ads & clk) then [1,0]
    else [0,1];

```

292074-33

Figure 19. iPLD610 ABEL Source File for 386 Bus Controller (Continued)



```

state [1,0]: "pipelined
  if (clk) then [0,1]
  else [1,0];

state [0,0]: "illegal
  goto [1,1];

test_vectors ([clk2,clk2a,clk,na,mio,wr,dc,pa31,timedly,buscyc] ->
  [iord,iowr,eprd,inta,iordy,recv]);

[c,c,h,h,h,h,h,h,h,h] -> [h,h,h,h,h,h,h,h]; "idle
[c,c,h,h,h,h,h,h,h,h] -> [h,h,h,h,h,h,h,h]; "idle
[c,c,h,h,h,h,h,h,h,h] -> [h,h,h,h,h,h,h,h]; "idle

[c,c,h,x,x,x,x,x,x,x] -> [h,h,h,h,h,h,h,h]; "preload buscyc
[c,c,h,h,h,h,1,1,h,h,1] -> [h,h,1,h,h,h,h,h]; "eprom read
[c,c,h,h,h,h,1,1,h,h,1] -> [h,h,1,h,h,h,h,h]; "eprom read
[c,c,h,h,h,h,1,1,h,h,1] -> [h,h,1,h,h,h,h,h]; "eprom read
[c,c,h,h,h,h,1,1,h,h,1] -> [h,h,1,h,h,h,h,h]; "eprom read
[c,c,h,h,h,h,1,1,h,h,1] -> [h,h,1,h,h,h,h,h]; "eprom read
[c,c,h,h,h,h,1,1,h,h,1] -> [h,h,1,h,h,h,h,h]; "eprom read
[c,c,h,h,h,h,h,h,h,h] -> [h,h,h,h,h,h,h,h]; "idle

[c,c,h,h,1,1,h,1,h,1] -> [1,h,h,h,h,h,h,h]; "io read
[c,c,h,h,1,1,h,1,h,1] -> [1,h,h,h,h,h,h,h]; "io read
[c,c,h,h,1,1,h,1,h,1] -> [1,h,h,h,h,h,h,h]; "io read
[c,c,h,h,1,1,h,1,h,1] -> [1,h,h,h,h,h,h,h]; "io read
[c,c,h,h,1,1,h,1,h,1] -> [1,h,h,h,h,h,h,h]; "io read
[c,c,h,h,h,h,h,h,h,h] -> [h,h,h,h,h,h,h,h]; "recovery
[c,c,h,h,h,h,h,h,h,h] -> [h,h,h,h,h,h,h,h]; "recovery
[c,c,h,h,h,h,h,h,h,h] -> [h,h,h,h,h,h,h,h]; "idle

[c,c,h,h,1,h,h,1,h,1] -> [h,1,h,h,h,h,h,h]; "io write
[c,c,h,h,1,h,h,1,h,1] -> [h,1,h,h,h,h,h,h]; "io write
[c,c,h,h,1,h,h,1,h,1] -> [h,1,h,h,h,h,h,h]; "io write
[c,c,h,h,1,h,h,1,h,1] -> [h,1,h,h,h,h,h,h]; "io write
[c,c,h,h,1,h,h,1,h,1] -> [h,h,h,h,h,h,h,h]; "recovery
[c,c,h,h,1,h,h,1,h,1] -> [h,h,h,h,h,h,h,h]; "recovery
[c,c,h,h,h,h,h,h,h,h] -> [h,h,h,h,h,h,h,h]; "idle

[c,c,h,h,1,1,1,1,h,1] -> [h,h,h,1,h,h,h]; "interrupt ack
[c,c,h,h,1,1,1,1,h,1] -> [h,h,h,1,h,h,h]; "interrupt ack
[c,c,h,h,1,1,1,1,h,1] -> [h,h,h,1,h,h,h]; "interrupt ack
[c,c,h,h,1,1,1,1,h,1] -> [h,h,h,1,h,h,h]; "interrupt ack
[c,c,h,h,1,1,1,1,h,1] -> [h,h,h,1,h,h,h]; "interrupt ack
[c,c,h,h,h,h,h,h,h,h] -> [h,h,h,h,h,h,h,h]; "recovery
[c,c,h,h,h,h,h,h,h,h] -> [h,h,h,h,h,h,h,h]; "recovery
[c,c,h,h,h,h,h,h,h,h] -> [h,h,h,h,h,h,h,h]; "idle

[c,c,h,h,h,h,h,1,1,h,1] -> [h,h,h,h,h,1,h,h]; "halt or shutdown

[c,c,h,h,h,h,h,h,h,h] -> [h,h,h,h,h,h,h,h]; "idle
[c,c,h,h,h,h,h,h,h,h] -> [h,h,h,h,h,h,h,h]; "idle

test_vectors ([clk2a, clk, ads, ready] -> [buscyc, pipecyc])

```

292074-34

Figure 19. iPLD610 ABEL Source File for 386 Bus Controller (Continued)

```

[p,h,l,l] -> [x,x];
[c,l,h,h] -> [x,h];
[c,h,h,l] -> [h,h];
[c,l,h,h] -> [h,h];
[c,h,l,h] -> [l,h];
[c,l,h,l] -> [l,h];
[c,h,h,l] -> [h,h];

"idle-busy-idle

[c,l,l,h] -> [h,h];
[c,h,l,h] -> [l,h];
[c,l,h,l] -> [l,h];
[c,h,l,l] -> [h,l];
[c,l,l,h] -> [h,l];
[c,h,h,h] -> [l,h];

"idle-busy-pipe-busy

[c,h,h,l] -> [h,h];
[c,l,h,h] -> [h,h];
[c,h,l,h] -> [l,h];
[c,l,l,h] -> [l,h];
[c,h,h,h] -> [l,h];
[c,l,h,h] -> [l,h];
[c,h,h,h] -> [l,h];
[c,l,h,c] -> [l,h];
[c,h,h,l] -> [h,h];
[c,l,h,h] -> [h,h];

"idle-busy-busy-idle

end pst060;

```

292074-35

3

Figure 19. IPLD610 ABEL Source File for 386 Bus Controller (Continued)

Device U1

- Reduced Equations for device U1:

```
!aleio := (!clk & !aleio # clk & !inta # clk & !iowr # clk & !iord);

!trioen := (!clk & !trioen
            # !buscyc & !trioen & iordy
            # clk & na & mio & pa31 & !buscyc & recv & trioen
            # clk & na & !mio & !pa31 & !buscyc & recv & trioen);

!iord := (!clk & recv & !iord & iowr & eprd & inta
          # recv & !iord & iowr & eprd & inta & iordy
          # clk & na & !mio & !wr & dc & !pa31 & !buscyc & recv & iowr &
          eprd & inta & iordy);

!iowr := (!clk & recv & iord & !iowr & eprd & inta & iordy
          # timedly & recv & iord & !iowr & eprd & inta & iordy
          # clk & na & !mio & wr & dc & !pa31 & !buscyc & recv & iord &
          iowr & eprd & inta & iordy);

!eprd := (!clk & recv & iord & iowr & !eprd & inta
          # recv & iord & iowr & !eprd & inta & iordy
          # clk & na & !wr & pa31 & !buscyc & recv & iord & iowr & inta
          & iordy);

!inta := (!clk & recv & iord & iowr & eprd & !inta
          # recv & iord & iowr & eprd & !inta & iordy
          # clk & na & !mio & !wr & !dc & !pa31 & !buscyc & recv & iord
          & iowr & eprd & iordy);

!iordy := (!clk & recv & iord & iowr & eprd & !iordy
           # clk & !timedly & recv & iord & iowr & eprd & !inta & iordy
           # !clk & recv & iowr & eprd & inta & !iordy
           # clk & !timedly & recv & !iord & iowr & eprd & inta & iordy
           # clk & !timedly & recv & iord & !iowr & eprd & inta & iordy
           # !clk & recv & iord & iowr & inta & !iordy
           # clk & !timedly & recv & iord & iowr & !eprd & inta & iordy
           # clk & na & mio & wr & !dc & !buscyc & recv & iord & iowr &
           eprd & inta & iordy);

!recv := (!clk & !recv & iord & iowr & eprd & inta & iordy
          # timedly & !recv & iord & iowr & eprd & inta & iordy
          # clk & recv & iord & iowr & eprd & !inta & !iordy
          # clk & recv & !iord & iowr & eprd & inta & !iordy
          # clk & !mio & recv & iowr & eprd & inta & !iordy);
```

292074-36

Figure 20. 386-PST .DOC File

Device U1

```
!buscyc := (clk & buscyc & !pipecyc
# !clk & !buscyc & pipecyc
# !buscyc & ready & pipecyc
# clk & buscyc & !ads);
```

```
!pipecyc := (!clk & buscyc & !pipecyc
# clk & !buscyc & !ads & !ready & pipecyc);
```

E0600

292074-37

clk2a	1	24	vcc
clk	2	23	mio
na	3	22	aleio
pipecyc	4	21	iordy
wr	5	20	trioen
dc	6	19	inta
pa31	7	18	eprd
timedly	8	17	iowr
buscyc	9	16	iord
	10	15	recv
ready	11	14	ads
gnd	12	13	clk2

292074-22

Figure 20. 386-PST .DOC File (Continued)

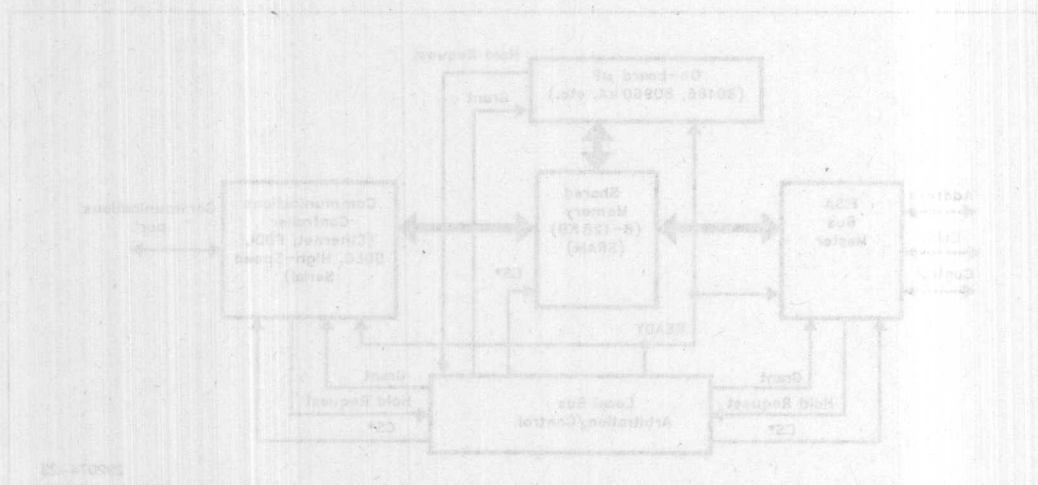


Figure 21. Intelligent ISA Communications Add-in Card



## 5.2 Shared Memory Arbitration/Bus Control

Sharing resources (usually memory) on a common bus is becoming more and more common in today's system designs. Not only are many systems, including personal computers, being implemented as multiprocessors, but peripheral controllers are becoming more intelligent and capable of controlling the bus by themselves. Thus, arbitration logic is required to determine which processor/controller currently has control of each shared resource. In addition, the designer may want to incorporate a variety of associated functions including:

- READY Logic (Bus Cycle Control)
- Address Pipelining Support
- Memory Burst Control
- Wait-State Generation
- I/O Chip Select Logic
- Bus Throttling Logic
- DRAM Control/Refresh Logic (if used instead of SRAM)
- EPROM Control Logic (if microprocessor code is not in SRAM)

The example to be discussed here is an intelligent EISA communications add-in card (see Figure 21). The arbitration logic must decide if the communications controller, on-board microprocessor or EISA bus controller has access to the on-board static RAM (SRAM). The on-board SRAM mainly acts as a high-speed data

buffer for the communications controller to off-load the EISA bus by providing block size transfers. The on-board microprocessor initializes the EISA control logic and communications controller and provides handling of local interrupts/error conditions. A block diagram of the system and required logic are provided by Figures 21 and 22, respectively. The "communications controller" may be implementing an ISDN, high-speed serial, Ethernet, FDDI or other communications link. Often these devices can themselves control the local bus and provide DMA capabilities to move data to and from memory. Also, many of these controllers have on-board data FIFOs that may necessitate a need for bursting data to/from the local memory.

Each of the three bus control-capable devices is assumed to have "Bus Request" and "Bus Grant" signals that are routed to the arbitration logic. The iPLD610 provides 8 product terms (in addition to separate OE and RESET product terms) in each macrocell, which allows for increased flexibility in choosing an arbitration scheme. Common schemes include fixed priority, rotating (or last granted, lowest priority), or First-Come, First-Serve. In this example a rotating priority is implemented to assure a balance of accesses between the devices, and to decrease the worst-case service time to help prevent data underflow/overflow. Figure 23 shows the state diagram of the 3-way bus arbitration logic. This method is easily expandable if more bus masters were present, as would be the case for DRAM Refresh requests and/or multiple communications controllers or in a multiprocessing system.

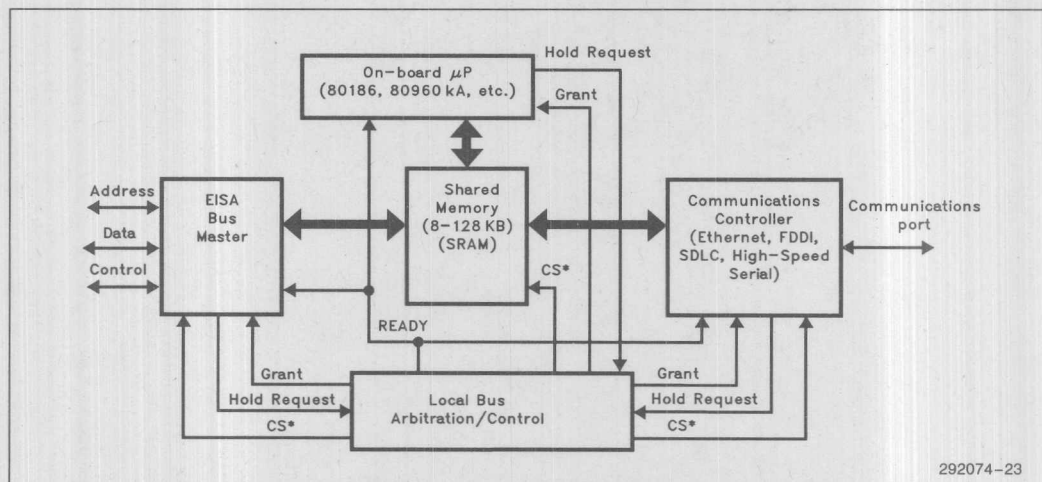


Figure 21. Intelligent EISA Communications Add-In Card

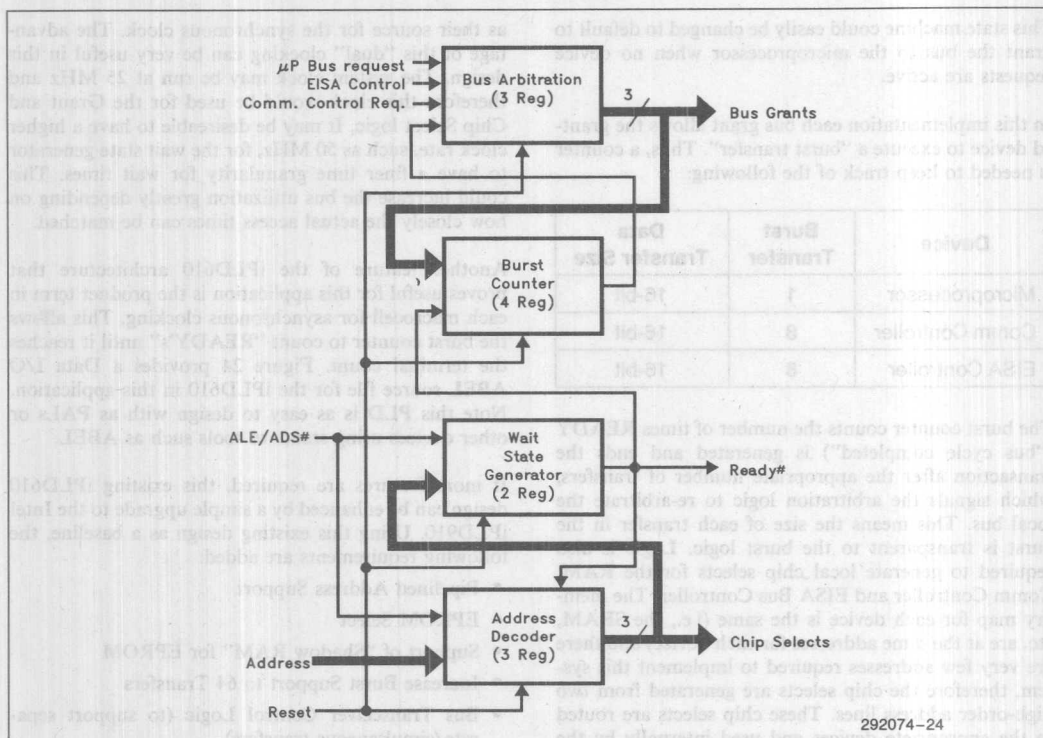


Figure 22. IPLD610 Shared Memory Arbitration/Bus Control Logic Implementation

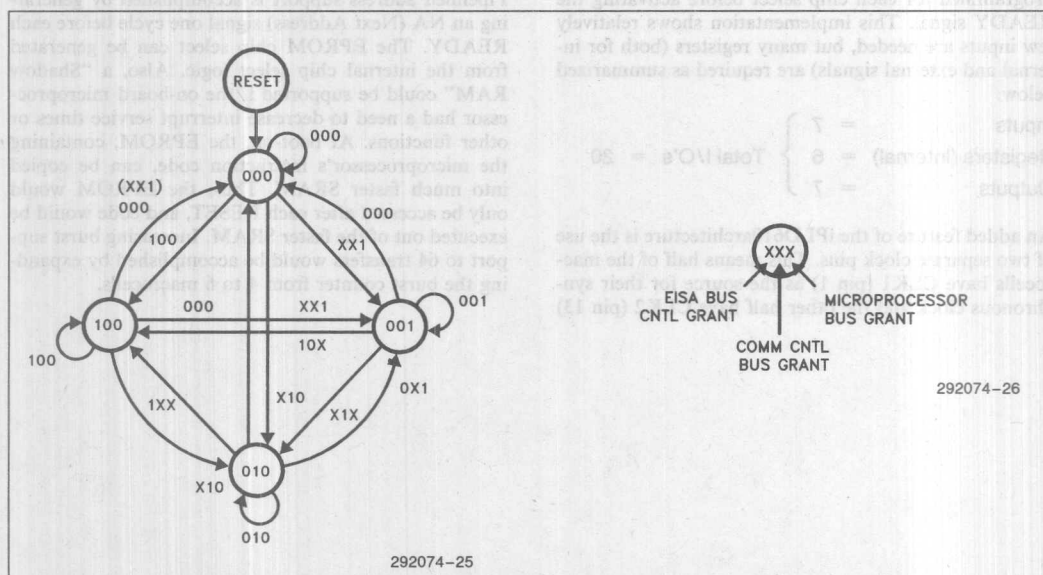


Figure 23. Bus Arbitration State Machine—Rotating Priority

This state machine could easily be changed to default to grant the bus to the microprocessor when no device requests are active.

In this implementation each bus grant allows the granted device to execute a "burst transfer". Thus, a counter is needed to keep track of the following:

Device	Burst Transfer	Data Transfer Size
Microprocessor	1	16-bit
Comm Controller	8	16-bit
EISA Controller	8	16-bit

The burst counter counts the number of times READY ("bus cycle completed") is generated and ends the transaction after the appropriate number of transfers, which signals the arbitration logic to re-arbitrate the local bus. This means the size of each transfer in the burst is transparent to the burst logic. Logic is also required to generate local chip selects for the RAM, Comm Controller and EISA Bus Controller. The memory map for each device is the same (i.e., the SRAM, etc. are at the same addresses for each device) and there are very few addresses required to implement this system, therefore the chip selects are generated from two high-order address lines. These chip selects are routed to the appropriate devices and used internally by the wait-state generator, which counts the number of cycles programmed for each chip select before activating the READY signal. This implementation shows relatively few inputs are needed, but many registers (both for internal and external signals) are required as summarized below:

Inputs	= 7	} Total I/O's = 20
Registers (Internal)	= 6	
Outputs	= 7	

An added feature of the iPLD610 architecture is the use of two separate clock pins. This means half of the macrocells have CLK1 (pin 1) as the source for their synchronous clock and the other half have CLK2 (pin 13)

as their source for the synchronous clock. The advantage of this "dual" clocking can be very useful in this design. The system clock may be run at 25 MHz and therefore this clock would be used for the Grant and Chip Select logic. It may be desirable to have a higher clock rate, such as 50 MHz, for the wait state generator to have a finer time granularity for wait times. This could increase the bus utilization greatly depending on how closely the actual access times can be matched.

Another feature of the iPLD610 architecture that proves useful for this application is the product term in each macrocell for asynchronous clocking. This allows the burst counter to count "READY's" until it reaches the terminal count. Figure 24 provides a Data I/O ABEL source file for the iPLD610 in this application. Note this PLD is as easy to design with as PALs or other devices using standard tools such as ABEL.

If more features are required, this existing iPLD610 design can be enhanced by a simple upgrade to the Intel iPLD910. Using this existing design as a baseline, the following requirements are added:

- Pipelined Address Support
- EPROM Select
- Support of "Shadow RAM" for EPROM
- Increase Burst Support to 64 Transfers
- Bus Transceiver Control Logic (to support separate/simultaneous transfers)

Pipelined address support is accomplished by generating an NA (Next Address) signal one cycle before each READY. The EPROM chip select can be generated from the internal chip select logic. Also, a "Shadow RAM" could be supported if the on-board microprocessor had a need to decrease interrupt service times or other functions. At boot-up, the EPROM, containing the microprocessor's instruction code, can be copied into much faster SRAM. Thus, the EPROM would only be accessed after each RESET, and code would be executed out of the faster SRAM. Increasing burst support to 64 transfers would be accomplished by expanding the burst counter from 4 to 6 macrocells.

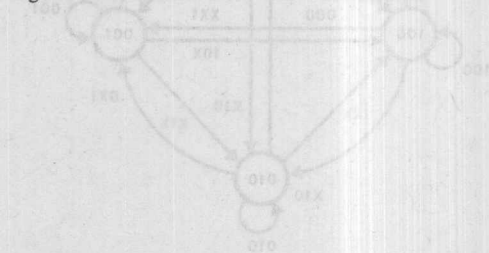


Figure 24. Bus Arbitration State Machine—Notating "Ready"

```

module arb_cntl          flag '-UL', '-r3'
title 'Arbitration/Bus control logic
John Casey      Intel Corp.      July 1990'

"THIS IS A SAMPLE FILE ONLY.  THIS CIRCUIT PROVIDES A GENERAL SOLUTION !!!

arblogic      device      'E0600';

"inputs
clk1          pin 1;      "33 MHz clock input
clk2          pin 13;     "33 MHz clock input
procreq       pin 2;      "microprocessor local bus request
commreq       pin 23;     "comm controller local bus request
eisareq       pin 11;     "on-board eisa controller local bus req.
addr1         pin 14;     "address input - used for chip select dec.
addr2         pin 3;      "address input - used for chip select dec.
ale_          pin 5;      "ale (or ads/) - represents valid addr.
reset         pin 22;     "board reset - used to set initial state

"outputs
procgrant     pin 6;      "processor local bus grant
commgrant     pin 7;      "communications controller local bus grant
eisagrnt      pin 8;      "on-board eisa controller local bus grant
ready_        pin 9;      "indicates end of current bus cycle
comms         pin 10;     "comm controller chip select
eisacs        pin 15;     "local eisa bus controller chip select
sramcs        pin 21;     "sram chip select

"buried functions
burst0        pin 16;     "part of burst counter logic
burst1        pin 17;     "part of burst counter logic
burst2        pin 18;     "part of burst counter logic
burstdone     pin 19;     "active at end of current burst count
wait0         pin 20;     "lower bit of wait state count logic,
                        "ready represents output of this logic
wait1         pin 4;      "upper bit of wait state count logic

"macrocell control
burst0, burst1, burst2, burstdone      istype 'feed_reg';
wait0, wait1                          istype 'feed_reg';

"busarb valid states (for the bus arbitration state machine)
s0 = `b000;      "no grant active
s1 = `b001;      "microprocessor grant active
s2 = `b010;      "comm controller grant active
s3 = `b100;      "eisa bus controller grant active

"wait state generator valid states
ws0 = `b000;      ws1 = `b010;      ws2 = `b100;      ws3 = `b111;

"burst state machine
bs0 = `b000;      bs4 = `b100;
bs1 = `b001;      bs5 = `b101;
bs2 = `b010;      bs6 = `b110;
bs3 = `b011;      bs7 = `b111;

```

292074-38

Figure 24. Sample IPLD610 ABEL Source File



## equations

```

commcs := !addr1 & !addr2;           "address mapping will vary
commcs.clk = !ale_;                  "ale is async clock

eisacs := !addr1 & addr2;
eisacs.clk = !ale_;

sramcs := addr1 & !addr2;
sramcs.clk = !ale_;

burstdone = burst0 & burst1 & burst2;    "terminal count completed

burst0.clk = ready_;                 "burst counter async clocked w/ ready
burst1.clk = ready_;
burst2.clk = ready_;

"Reset signal sets all registers low
procgrant.re = reset;   commgrant.re = reset;   eisagrant.re = reset;
commcs.re = reset;      eisacs.re = reset;      sramcs.re = reset;
burst0.re = reset;      burst1.re = reset;      burst2.re = reset;
wait0.re = reset;       wait1.re = reset;

```

"busarb state machine uses a rotating (last grant, lowest priority) scheme  
 "which allows bus accesses to be balanced. If no request then returns to  
 "state 000 which allows microprocessor to be #1 priority (this is due to  
 "requirement to quickly service interrupts.

```

state_diagram [procgrant, commgrant, eisagrant]
State s0: if (burstdone & procreq) then s1 else s0;
          if (burstdone & commreq & !procreq) then s2 else s0;
          if (burstdone & eisareq & !commreq & !procreq) then s3 else s0;

State s1: if (burstdone & commreq) then s2 else s0;
          if (burstdone & eisareq & !commreq) then s3 else s0;
          if (burstdone & procreq & !eisareq & !commreq) then s1 else s0;

State s2: if (burstdone & eisareq) then s3 else s0;
          if (burstdone & procreq & !eisareq) then s1 else s0;
          if (burstdone & commreq & !procreq & !eisareq) then s3 else s0;

State s3: if (burstdone & procreq) then s1 else s0;
          if (burstdone & commreq & !procreq) then s2 else s0;
          if (burstdone & eisareq & !commreq & !procreq) then s3 else s0;

```

"This wait state generator can support accesses up to 4 clocks long.  
 "The EISA controller and the comm controller each take 4 cycles (2 wait  
 "states) and the SRAM takes two cycles (zero wait states).

```

state_diagram [wait0, wait1, ready_]

State ws0:   if (sramcs) then ws3 else ws0;
             if (commcs # eisacs) then ws1 else ws0;

State ws1:   goto ws2;

```

292074-39

Figure 24. Sample IPLD610 ABEL Source File (Continued)

```
State ws2:      goto ws3;
```

```
State ws3:      goto ws0;
```

"the burst logic state machine has a programmed burst length for each bus grant. These are 8 transfers for the comm controller (fifo) and EISA bus controller and one transfer for the microprocessor. When terminal count is reached BURSTDONE will go active, indicating to the bus arbitration logic that it should re-arbitrate the bus. At each state the bus grant is checked to make sure it is still active (i.e. that a full FIFO transfer is necessary).

```
state_diagram [burst0, burst1, burst2]
state bs0: if (procgrant) then bs7 else bs0;
           if (commgrant & !eisagrnt) then bs1 else bs0;
state bs1: if (!commgrant & !eisagrnt) then bs7 else bs2;
state bs2: if (!commgrant & !eisagrnt) then bs7 else bs3;
state bs3: if (!commgrant & !eisagrnt) then bs7 else bs4;
state bs4: if (!commgrant & !eisagrnt) then bs7 else bs5;
state bs5: if (!commgrant & !eisagrnt) then bs7 else bs6;
state bs6: goto bs7;
state bs7: goto bs0;
```

```
"      test_vectors
"      .
"      .
"      .
```

```
end arb_cntl;
```

292074-40

Figure 24. Sample IPLD610 ABEL Source File (Continued)

### 5.3 High-Speed Custom Control/ Status Register

In many applications there exists a need for a high-speed dedicated control and/or status register. In multiprocessing systems there may be a need for a system control register; in a communications controller there may be a requirement for a system configuration control register. Figure 25 shows this general application idea. There is a frequent requirement in system add-on card and adapter designs for a Read/Write register that also provides discrete status inputs and control outputs.

The destinations of these discrete signals can include a wide variety of functions that need to be under software control. The functions provided by this register can include enable signals, transceiver/mux control and it

can also include status indications such as a communications error. The design, configuration and purpose of each bit of this register is individually selectable, which leads into a wide variety of applications.

The iPLD610 is useful in this application due to these key architectural features:

- 16 macrocells (8 for data bus, 8 for control outputs/inputs)
- separate product term for RESET (to set known power up state)
- four dedicated inputs (necessary for register control)
- pin/register feedback capabilities and OE control (allows specialized I/O capabilities to be implemented)

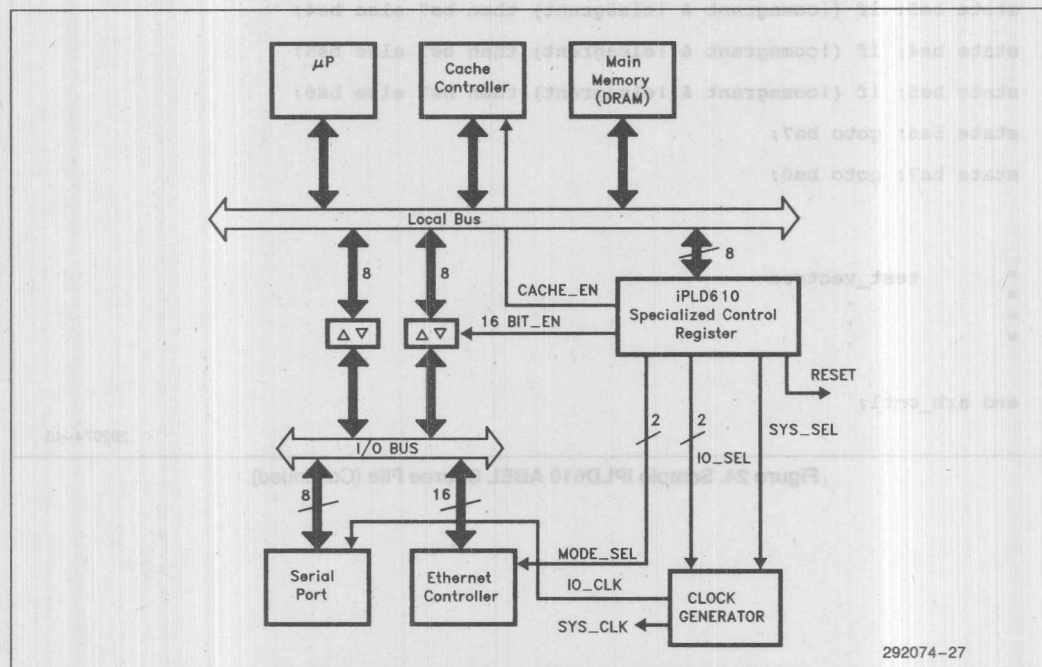


Figure 25. Typical System Requiring Specialized Register for Configuration Control

As shown in Figures 26 and 27, the iPLD610 uses these features to implement each bit of this control register. Not only is the architecture of the iPLD610 a perfect match for this application, but the performance level is so high (83.3 MHz register speed) that it can meet the zero-wait state requirements for most microprocessors. This provides very high-speed register access via Read-Modify-Write operations.

Some applications may benefit from the power-down mode of the iPLD610. If the register implemented is required only periodically, such as at power-on or after system resets, the power-down feature of the iPLD610 can be programmed to allow power consumption to be in the 20  $\mu$ A (typical) range. Decreasing power consumption can have a positive impact on system reliability in addition to decreasing power supply and cooling requirements.

A sample design file is included in Figure 28. This demonstrates the ease of using the advanced architectural capabilities of the 85C060. Though this example uses the 85C060, the iPLD610 can be used by simply choosing the 85C060 from the IPLS II menu. This design file is written in the Intel ADF (Advanced Design File) format and is compiled by the iPLS II Development Tools. Since the creation of this example, Intel has updated its tools support. Intel's newest tool, PLDshell Plus, is a free tool that supports Intel's entire PLD and FPGA product lines. (See the PLD Handbook or call the EPLD hotline, 800-323-EPLD, for more information on Intel development tools.) Figure 27 provides details on the actual logic implementation of each of the eight register/control output cells. Each of these cells requires two macrocells of the 85C060. One macrocell forms the register portion and provides the control signal and the other provides the data bus interface (with output enable controlled).

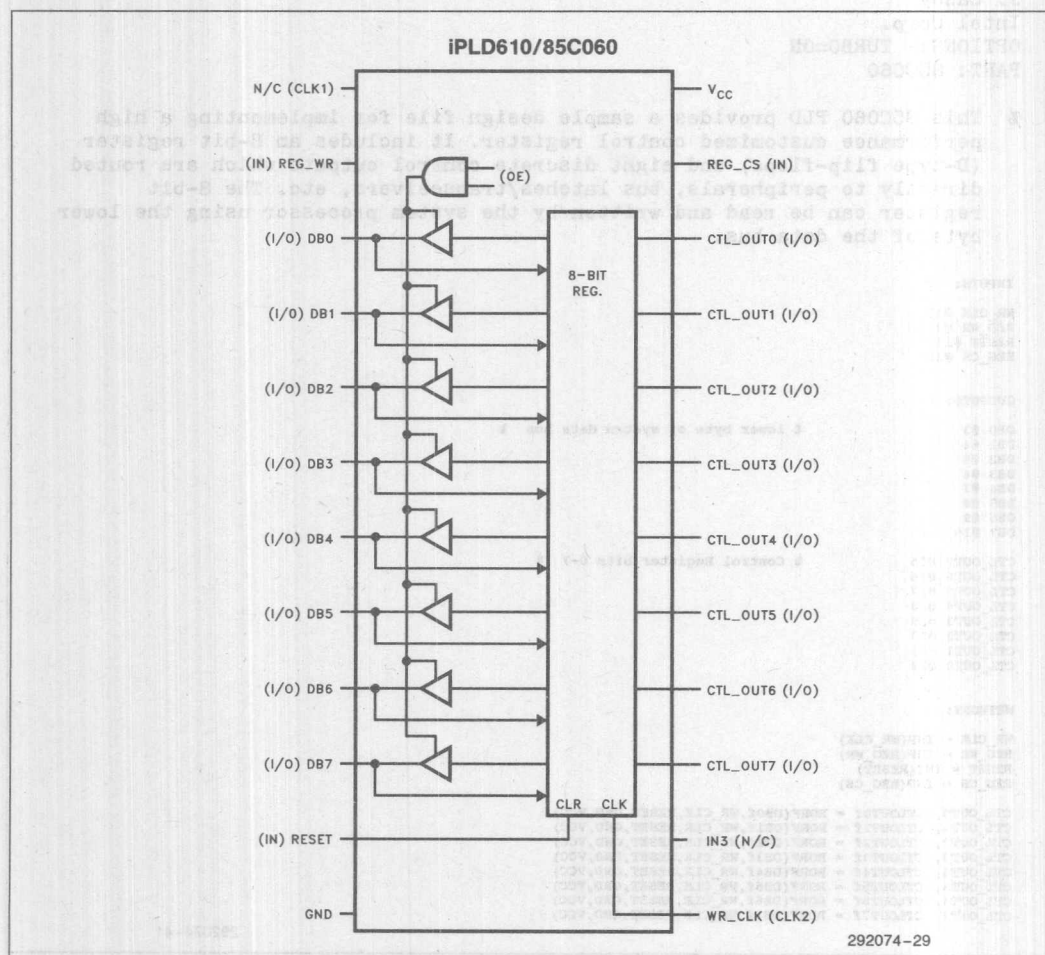


Figure 26. Device Implementation of Control/Status Register



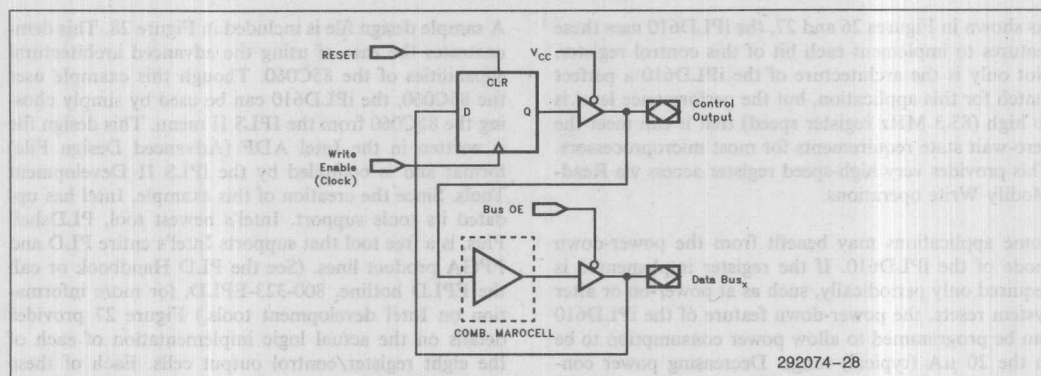


Figure 27. iPLD610/85C060 Control Register Implementation (1-Bit)

J. Casey  
Intel Corp.  
OPTIONS: TURBO=ON  
PART: 85C060

% This 85C060 PLD provides a sample design file for implementing a high performance customized control register. It includes an 8-bit register (D-type flip-flops) and eight discrete control outputs which are routed directly to peripherals, bus latches/transceivers, etc. The 8-bit register can be read and written by the system processor using the lower byte of the data bus.

#### INPUTS:

WR\_CLK @13  
REG\_WR @2  
RESET @11  
REG\_CS @23

#### OUTPUTS:

DB0 @3                   % lower byte of system data bus %  
DB1 @4  
DB2 @5  
DB3 @6  
DB4 @7  
DB5 @8  
DB6 @9  
DB7 @10

CTL\_OUT7 @15           % Control Register bits 0-7 %  
CTL\_OUT6 @16  
CTL\_OUT5 @17  
CTL\_OUT4 @18  
CTL\_OUT3 @19  
CTL\_OUT2 @20  
CTL\_OUT1 @21  
CTL\_OUT0 @22

#### NETWORK:

WR\_CLK = INP(WR\_CLK)  
REG\_WR = INP(REG\_WR)  
RESET = INP(RESET)  
REG\_CS = INP(REG\_CS)

CTL\_OUT0, CTL\_OUT0f = RORF(DB0f, WR\_CLK, RESET, GND, VCC)  
CTL\_OUT1, CTL\_OUT1f = RORF(DB1f, WR\_CLK, RESET, GND, VCC)  
CTL\_OUT2, CTL\_OUT2f = RORF(DB2f, WR\_CLK, RESET, GND, VCC)  
CTL\_OUT3, CTL\_OUT3f = RORF(DB3f, WR\_CLK, RESET, GND, VCC)  
CTL\_OUT4, CTL\_OUT4f = RORF(DB4f, WR\_CLK, RESET, GND, VCC)  
CTL\_OUT5, CTL\_OUT5f = RORF(DB5f, WR\_CLK, RESET, GND, VCC)  
CTL\_OUT6, CTL\_OUT6f = RORF(DB6f, WR\_CLK, RESET, GND, VCC)  
CTL\_OUT7, CTL\_OUT7f = RORF(DB7f, WR\_CLK, RESET, GND, VCC)

292074-41

Figure 28. iPLS II Source File for 85C060 Custom Register Design

```

DB0, DB0f = COIF(iDB0,REG_OE)
DB1, DB1f = COIF(iDB1,REG_OE)
DB2, DB2f = COIF(iDB2,REG_OE)
DB3, DB3f = COIF(iDB3,REG_OE)
DB4, DB4f = COIF(iDB4,REG_OE)
DB5, DB5f = COIF(iDB5,REG_OE)
DB6, DB6f = COIF(iDB6,REG_OE)
DB7, DB7f = COIF(iDB7,REG_OE)

```

#### EQUATIONS:

```

iDB0 = CTLOUT0f;
iDB1 = CTLOUT1f;
iDB2 = CTLOUT2f;
iDB3 = CTLOUT3f;
iDB4 = CTLOUT4f;
iDB5 = CTLOUT5f;
iDB6 = CTLOUT6f;
iDB7 = CTLOUT7f;

```

```
REG_OE = REG_CS * !REG_WR;
```

```
END$
```

292074-42

Figure 28. iPLS II Source File for 85C060 Custom Register Design (Continued)

A .RPT (Report) file was generated by the Intel iPLS logic compiler. A JEDEC for the 85C060 was also generated by the software. This .RPT file, included in Figure 29, shows the macrocell/p-term usage and device pin-out as the device was implemented.

This application demonstrates the effective combination of architecture and performance provided by the Intel 85C060  $\mu$ PLD. In essence, the performance increase over existing devices with identical architecture has opened more doors for additional uses. It is now clear how this device can be used in high-speed applications—even when closely coupled to the microprocessor.

INTEL Logic Optimizing Compiler Utilization Report  
iPLS II FIT Version 2.2 Beta3 Level 4.0i 9/7/88

a:CONTROL.rpt

\*\*\*\*\* Design implemented successfully

J. Casey  
Intel Corp.

OPTIONS: TURBO=ON

#### 85C060

Gnd	1	24	Vcc
REG_WR	2	23	REG_CS
DB0	3	22	CTL_OUT0
DB1	4	21	CTL_OUT1
DB2	5	20	CTL_OUT2
DB3	6	19	CTL_OUT3
DB4	7	18	CTL_OUT4
DB5	8	17	CTL_OUT5
DB6	9	16	CTL_OUT6
DB7	10	15	CTL_OUT7
RESET	11	14	Gnd
Gnd	12	13	WR_CLK

292074-30

Figure 29. iPLS II Report File For The 85C060 Custom Register Design

**\*\*OUTPUTS\*\***

Name	Pin	Resource	MCell	PTerms	Sync Clock
DB0	3	COIF	9	1/ 8	-
DB1	4	COIF	10	1/ 8	-
DB2	5	COIF	11	1/ 8	-
DB3	6	COIF	12	1/ 8	-
DB4	7	COIF	13	1/ 8	-
DB5	8	COIF	14	1/ 8	-
DB6	9	COIF	15	1/ 8	-
DB7	10	COIF	16	1/ 8	-
CTL_OUT7	15	RORF	8	1/ 8	WR_CLK
CTL_OUT6	16	RORF	7	1/ 8	WR_CLK
CTL_OUT5	17	RORF	6	1/ 8	WR_CLK
CTL_OUT4	18	RORF	5	1/ 8	WR_CLK
CTL_OUT3	19	RORF	4	1/ 8	WR_CLK
CTL_OUT2	20	RORF	3	1/ 8	WR_CLK
CTL_OUT1	21	RORF	2	1/ 8	WR_CLK
CTL_OUT0	22	RORF	1	1/ 8	WR_CLK

**\*\*INPUTS\*\***

Name	Pin	Resource	MCell	PTerms	Sync Clock
WR_CLK	13	CKR	-	-	-
REG_WR	2	INP	-	-	-
RESET	11	INP	-	-	-
REG_CS	23	INP	-	-	-

**\*\*UNUSED RESOURCES\*\***

Name	Pin	Resource	MCell	PTerms
-	14	INPUT	-	-

**\*\*PART UTILIZATION\*\***

16/16      Macrocells (100%), 12% of used Pterms Filled  
 3/ 4      Input Pins (75%)  
          Pterms Used 12%

**Figure 29. iPLS II Report File For The 85C060 Custom Register Design (Continued)**

## Macrocell Interconnection Cross Reference

FEEDBACKS:			M M M M M M M M	M M M M M M M M
			0 0 0 0 0 0 0 0	0 1 1 1 1 1 1 1
			1 2 3 4 5 6 7 8	9 0 1 2 3 4 5 6
CTL_OUT0	. RORF @M1 ->	.	.	@22
CTL_OUT1	. RORF @M2 ->	.	.	@21
CTL_OUT2	. RORF @M3 ->	.	.	@20
CTL_OUT3	. RORF @M4 ->	.	.	@19
CTL_OUT4	. RORF @M5 ->	.	.	@18
CTL_OUT5	. RORF @M6 ->	.	.	@17
CTL_OUT6	. RORF @M7 ->	.	.	@16
CTL_OUT7	. RORF @M8 ->	.	.	@15
DB0	..... COIF @M9 ->	*	.	@3
DB1	..... COIF @M10->	.	*	@4
DB2	..... COIF @M11->	.	.	@5
DB3	..... COIF @M12->	.	.	@6
DB4	..... COIF @M13->	.	.	@7
DB5	..... COIF @M14->	.	.	@8
DB6	..... COIF @M15->	.	.	@9
DB7	..... COIF @M16->	.	.	@10
INPUTS:				
REG_WR	... INP @2 ->	.	*	
RESET	... INP @11 ->	*	.	
WR_CLK	... CKR @13 ->	*	.	
REG_CS	... INP @23 ->	.	*	
		C C C C C C C C	D D D D D D D D	
		T T T T T T T T	B B B B B B B B	
		L L L L L L L L	0 1 2 3 4 5 6 7	
		0 0 0 0 0 0 0 0		
		U U U U U U U U		
		T T T T T T T T		
		0 1 2 3 4 5 6 7		

. = not connected                      x = no connection possible  
 \* = signal feeds cell                ? = error, unable to fit

292074-43

Figure 29. iPLS II Report File For The 85C060 Custom Register Design (Continued)



## 6.0 iPLD610 PROGRAMMING/ DEVELOPMENT SUPPORT

Design development and device programming support are important issues for PLDs because the silicon is useless without them. Design development tools such as Intel's PLDshell Plus and Data I/O's ABEL are required to convert state machine/boolean equation entries into the required device JEDEC file. The device programmer is then required to program each cell/fuse to configure the device to the user's needs.

Since devices with architectures identical to the iPLD610 have been available for a number of years, there is an existing level of support for this PLD already in existence. As mentioned earlier, the architecture, pin-out, and JEDEC map for the iPLD610 is identical to the Intel 5C060, Altera EP600, EP610, and EP630, AMD PALCE630, and TI EP610 and EP630. Therefore, performance of existing designs using these devices can be improved by replacing these parts using the existing design/JEDEC files.

Full logic compilation and functional simulation for the iPLD610 is supported by PLDshell Plus software. The GUPI LOGICIID provides programming support on Intel programmers.

PLDshell Plus design software is Intel's new, user-friendly design tool for  $\mu$ PLD design. PLDshell Plus allows users to incorporate their preferred text editor, programming software, and additional design tools into a easy-to-use, menued design environment that includes Intel's PLDasm logic compiler and simulation software along with disassembly, conversion, and translation utilities. The PLDasm compiler and simulator software accepts industry standard PDS source files that express designs as Boolean equations, truth tables, or state machines. On-line help, datasheet briefs, technical notes and error message information, along with waveform viewing/printing capability make the design task as easy as possible. PLDshell Plus software is available from Intel Literature channels or from your local Intel sales representative.

Tools that support schematic capture and timing simulation of the iPLD610 are available. Support under iPLS II is still available. Please refer to the "Development Tools" section of the Programmable Logic handbook.

The iPLD610 is supported by third-party logic compilers such as ABEL\*, CUPL\*, PLDesigner\*, Log/IC, etc. Programming support is provided by third-party programmer companies such as Data I/O, Logical Devices, STAG, etc. Please refer to the "Third-Party Support" lists in the *Programmable Logic* handbook for complete information and vendor contacts.

The support summary shows an emphasis by Intel to provide timing/simulation models from Viewlogic and Quadtree for customers performing device/board simulations. The models are provided on most popular design platforms. The support summary reflects Intel's close relationship with key support tool vendors (which includes others not shown here) to provide iPLD610 customers design support with the format and package with which they are most familiar. These support tool vendors, such as Data I/O and Logical Devices, provide products in addition to the Intel development products. Intel provides complete design development support as detailed in Figure 5.2. Designs can be entered in several ways:

1. Schematic Capture using IPLDview-286—this Viewlogic Workview derivative allows gate/74xx macro level entry and supports design simulation (functional or timing).
2. State Machine entry using iSTATE—this Intel tool translates SMF (State Machine Format) designs into an ADF file.
3. Boolean Equation entry using PLDshell Plus—this logic compiler/minimizer generates device JEDEC/Report files from an ADF source file.

Based on the variety of offerings, users can pick the tool(s) that best fits their development environment. Also, programming support is available with Intel's GUPI adapters, which provide a quick and efficient method of device programming.

## 7.0 UPGRADING TO THE iPLD610

Because of the performance/architecture combination provided by the iPLD610 it may be desirable to upgrade an existing design for many reasons. Upgrading from 5C060 or EP6x0 devices can provide quick performance increases. Upgrading from 22V10 or 20RA10 devices may be done for performance, architecture, heat or power consumption reasons and may include the need for the standby power mode. Conversion of each of the devices mentioned here follows.

### UPGRADE FROM 5C060

The Intel 5C060 PLD is pin- and JEDEC-compatible with the iPLD610. This allows existing 5C060 JEDEC files to be programmed directly into iPLD610. Therefore, designs running on 45 ns–55 ns  $t_{PD}$  performance levels can instantly be upgraded to 12 ns–25 ns  $t_{PD}$  performance levels provided by the iPLD610. Intel development tools including PLDshell Plus and GUPI Logic IID programmer adapter support both Intel PLDs. Most non-Intel design development tools and device programmers also support both Intel devices.

### UPGRADE FROM ALTERA/TI/AMD EP6x0

As discussed in Section 3, the A.C. and D.C. specifications of the iPLD610 meet or exceed those of competitive devices from Altera, TI and AMD. The list of devices which can be upgraded to the Intel iPLD610 includes:

- Altera EP600
- Altera EP610
- Altera EP610A
- Altera EP630
- TI EP610
- TI EP630
- AMD PALCE 610
- AMD PALCE 630
- Cypress EP610

Just as with the Intel 5C060, an upgrade of one of these devices can be accomplished by using existing JEDEC files from one of these devices to program directly onto an iPLD610. Also, Data I/O ABEL, Logical Devices CUPL and other design tools support the iPLD610. The only difference of note is the lack of a programmable standby current mode in the AMD device. Again, due to complete architecture compatibility the iPLD610 provides a quick and easy performance increase to any design using any of the above devices.

Due to differences in device programming parameters (such as programming voltages and pulse width requirements) all device programmers may not support the iPLD610 unless explicitly stated. Also, device simulation models are not interchangeable between devices due to timing differences.

### UPGRADE FROM 22V10

Many designers requiring more architectural features than standard PAL devices offer have turned to the 22V10. Compared to standard PALs, the 22V10 offers additional outputs and product terms. However, the iPLD610 also has features not found in PAL devices. Since there are many differences between the iPLD610 and 22V10, the analysis will revolve first around device pin-out and then, cover performance, architecture, clocking options and D.C. specification issues.

### Device Pinout

Table 9 shows the device pinouts of the iPLD610 and the 22V10 and highlights the differences. The iPLD610 has dedicated clock inputs at pins 1 and 13 while both of these can be inputs on the 22V10. The iPLD610 also has dedicated inputs at pins 14 and 23, while these are

Table 9. iPLD610/22V10 Pin-Out Comparison

DIP Pin #	iPLD610	22V10	Notes
1	CLK1	CLK/INPUT	Clock only on iPLD610
2	INPUT	INPUT	
3	I/O	INPUT	Input only on 22V10
4	I/O	INPUT	Input only on 22V10
5	I/O	INPUT	Input only on 22V10
6	I/O	INPUT	Input only on 22V10
7	I/O	INPUT	Input only on 22V10
8	I/O	INPUT	Input only on 22V10
9	I/O	INPUT	Input only on 22V10
10	I/O	INPUT	Input only on 22V10
11	INPUT	INPUT	
12	GND	GND	
13	CLK2	INPUT	Clock only on iPLD610
14	INPUT	I/O	Input only on iPLD610
15	I/O	I/O	
16	I/O	I/O	
17	I/O	I/O	
18	I/O	I/O	
19	I/O	I/O	
20	I/O	I/O	
21	I/O	I/O	
22	I/O	I/O	
23	INPUT	I/O	Input only on iPLD610
24	V <sub>CC</sub>	V <sub>CC</sub>	

I/O pins on the 22V10. However, there are 8 input-only pins (3–10) on the 22V10 that are I/O pins on the iPLD610. This is the main architectural advantage provided by the iPLD610. Table 10 shows a summary of performance (A.C. specifications of the iPLD610) compared to the 22V10–15. This includes combinational logic speed ( $t_{PD}$ ), register performance ( $t_{CO1}$  and  $t_{SU}$ ), and maximum register speed ( $F_{MAX}$ ).

## ARCHITECTURE

A comparison of architectures shows that the iPLD610 has 6 more I/O pins (macrocells) than the 22V10, even though both are 24-pin devices. The main advantages of the 22V10 architecture are higher number of p-terms/macrocell (up to 16 for some macrocells) and capability of up to 22 inputs (compared to 20 for the iPLD610).

The number of p-terms (product terms or AND terms) required by any design is often greatly affected by the register type used. The 22V10 offers only D-type flip-flops, while the iPLD610 offers D, T, JK or RS register types. Register type selection is done on a macrocell-by-macrocell basis, so each specific function with the iPLD610 can be better optimized. Figure 30 shows an example of how register type selection can affect the number of p-terms required. The example shown is an MSB of a 4-bit counter which requires 5 p-terms using a D-type flip-flop, but only one p-term if a T-type flip-flop is used. By using the register type selection capability of the iPLD610 the designer can further minimize logic requirements.

Table 10. iPLD610/22V10 Specification Comparison

Area of Comparison	Feature	iPLD610-15	AMD 22V10-15	Lattice GAL 22V10-15
Performance	$t_{PD}$	15 ns	15 ns	15 ns
	$F_{CNT}$	50 MHz	50 MHz	50 MHz
	$F_{MAX}$	66 MHz	50 MHz	62.5 MHz
	$t_{CO1}$	8 ns	10	8 ns
	$t_{SU}$	12 ns	10	12 ns
Architecture	Macrocells	16	10	10
	Outputs	16	10	10
	Max Inputs	20	22	22
	Prog. Security Cell	Yes	Yes	Yes
	Ave P-Terms/Macrocell	8	12	12
	Individual Macrocell OE	Yes	Yes	Yes
	Reg Clock Options	3	1	1
	Indiv. Macrocell Clear	Yes	Yes	No
	Invert Control	Yes	Yes	Yes
	Reg. Power Up State	Low	High	High
	Register Types	D/T/RS/JK	D	D
D.C. Specifications	$I_{CC}(\max)$	105 mA	180 mA	130*
	Power-Down Mode	Yes	No	No
	Standby Current	<100 $\mu A$	N/A	N/A
	$I_{OL}(\max)$	12 mA	16 mA	16 mA

\* $F_{TOGGLE} = 15 \text{ MHz}$ 

### Clocking Options

One other major architectural improvement provided by the iPLD610 is in the area of logic clocking options. Not only does the iPLD610 have two dedicated clock inputs (allowing separate synchronous clocking of macrocells), but each macrocell has a p-term dedicated as an asynchronous clock option.

### D.C. Specifications

The final area of comparison between these two devices is D.C. specifications. The advantage in maximum supply current ( $I_{CC}$ ) has benefits in many areas including power supply selection, cooling requirements and system reliability. The iPLD610 also offers a programmable standby current option which allows  $I_{CC}$  values to drop to 20  $\mu A$  (typical) when in the standby mode. This can be useful if working with a "power budget" such as in laptop PC or Microchannel add-in card design. The difference in  $I_{OL}$  capabilities is very small (16 mA for the 22V10 versus 12 mA for the iPLD610).



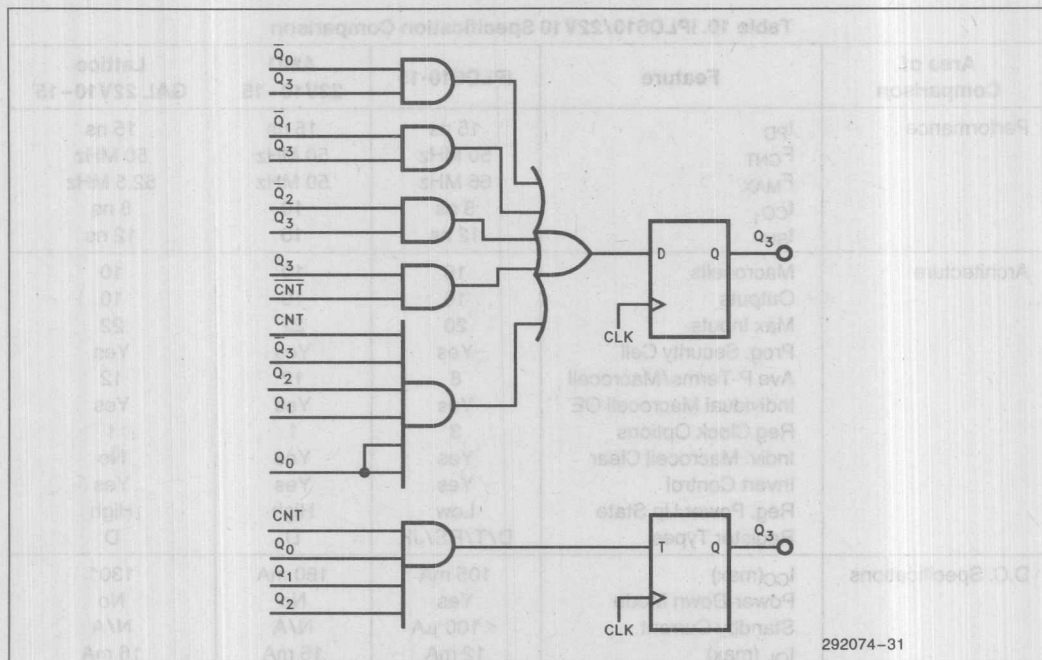


Figure 30. Product Term Requirements for D-Type and T-Type Flip-Flops (MSB of 4-Bit Counter)

### 22V10 to iPLD610 Conversion

#### NOTE:

There are NO performance related issues for this comparison due to design of iPLD610. No issues with registered operation ( $F_{MAX}$ ,  $t_{CO}$ ); no issues with combinatorial logic ( $t_{PD}$ ).

There are several ways to replace a 22V10. A "pin-for-pin" replacement means the iPLD610 that is upgrading the 22V10 can be placed into an existing 22V10 socket and provide complete compatibility. A "functional" replacement is one in which the iPLD610 can replace the 22V10, but one or more of the pin numbers need to be changed. A functional replacement will only be an issue if the board is already made; if the upgrade is made at the design phase there is very little impact.

1. Determine if a pin for pin replacement can be made:
  - a) Are there 8 or less p-terms used for each 22V10 output?
  - b) Is pin 1 used as a Clock (rather than an Input)?
  - c) Is pin 13 unused (i.e., is this input pin unused on the 22V10)?
  - d) Is the global register preset feature unused?

- e) Is the output drive ( $I_{OL}$ ) requirement for each output 12 mA or less?

#### NOTE:

Check for need of clock input at Pin 13 of the iPLD610.

If YES to all above, then pin-for-pin replacement looks good!

2. Determine if functional replacement can be made:
  - a) Is the total number of inputs and outputs 20 or less?
    - If NO, can the number be reduced via iPLD610 register type selection, inversion control (DeMorgan's logic implementation) and/or use of async clock capabilities?
  - c) Is the output drive ( $I_{OL}$ ) requirement for each output 12 mA or less?
    - If NO, is there a spare iPLD610 output which can be utilized to duplicate this output and double the output drive capability?

If YES to questions above, then a functional replacement looks good!

3. In some cases the 5AC312 can also upgrade 22V10 sockets due to it's pin-for-pin compatibility, and superset architecture, although it's performance is not equal to that of the faster 22V10s.

4. In other cases the 22V10 and surrounding PLDs and/or 74xxx logic can be integrated into one of Intel's larger high-performance PLDs (such as the iPLD910). This is always an option if there are power, heat, or board space concerns.

Call The Intel EPLD Hotline for Conversion Assistance  
1-800-323-EPLD

### UPGRADE FROM 20RA10

The 20RA10 presents designers requiring implementation of an asynchronous state machine or other asynchronous registered logic with a viable solution. The Intel iPLD610 provides these designers with another alternative. In many cases, a pin-compatible upgrade can be made due to architecture and performance features of the Intel iPLD610.

### Device Pin-Out

Table 11 compares the device pinouts of the iPLD610 and the Lattice 20RA10. Both have up to 20 inputs, both are 24 pin devices (DIP package), and both have dedicated functions implemented at pins 1 and 13. The advantage provided by the iPLD610 are the 16 I/Os (macrocells), compared to only 10 for the 20RA10. This allows designers the flexibility to implement larger asynchronous state machines and/or additional synchronous or combinatorial logic.

### Spec Comparison

A data sheet comparison of the Intel iPLD610 and Lattice 20RA10-15 is summarized in Table 12. The performance of the iPLD610 is superior for both combinatorial logic (t<sub>pd</sub>) and asynchronous register performance (F<sub>ACNT</sub>, F<sub>AMAX</sub>, t<sub>ACO1</sub>, t<sub>ASU</sub>) as well as synchronous logic (which is not supported by the 20RA10). There are significant differences in register setup time that could be critical to a designer required to meet specifications of microprocessors or other system peripherals (such as an output valid delay spec).

Table 11. iPLD610/20RA10 Pin-Out Comparison

DIP Pin #	iPLD610	22V10	Notes
1	CLK1	PRELOAD	Dedicated on Each
2	INPUT	INPUT	
3	I/O	INPUT	Input only on 20RA10
4	I/O	INPUT	Input only on 20RA10
5	I/O	INPUT	Input only on 20RA10
6	I/O	INPUT	Input only on 20RA10
7	I/O	INPUT	Input only on 20RA10
8	I/O	INPUT	Input only on 20RA10
9	I/O	INPUT	Input only on 20RA10
10	I/O	INPUT	Input only on 20RA10
11	INPUT	INPUT	
12	GND	GND	
13	CLK2	OE	Dedicated on Each
14	INPUT	I/O	Input only on iPLD610
15	I/O	I/O	
16	I/O	I/O	
17	I/O	I/O	
18	I/O	I/O	
19	I/O	I/O	
20	I/O	I/O	
21	I/O	I/O	
22	I/O	I/O	
23	INPUT	I/O	Input only on iPLD610
24	V <sub>CC</sub>	V <sub>CC</sub>	

Table 12. iPLD610/20RA10 Specification Comparison Summary

Area of Comparison	Feature	iPLD610-10	Lattice 20RA10-15
Performance	$t_{PD}$	10 ns	12 ns
	$F_{ACNT1}$	71.4 MHz	62.5 MHz
	$F_{AMAX}$	100 MHz	71.4 MHz
	$t_{ACO1}$	12 ns	12 ns
	$t_{ASU}$	2 ns	7 ns
Architecture	DIP Pin Count	24	24
	Macrocells	16	10
	Outputs (I/Os)	16	10
	Inputs (max)	20	20
	Prog. Security Cell	Yes	No
	Total P-Terms/Macrocell	10	8
	Indiv. Macrocell OE	Yes	No
	Reg. Preload	No	Yes
	Invert Control	Yes	Yes
	Reg. Types	D/T/JK/RS	D
	Clock Options	Sync/Async	Async Only
	Async Preset	No	Yes
	Async Clear	Yes	Yes
D.C. Specs	$I_{CC}$ (max)	105 mA@1 MHz	100 mA@15 MHz
	Power Down Mode	Yes	No
	Standby $I_{CC}$	150 $\mu$ A (max)	N/A
	$I_{OL}$ (max)	12 mA	8 mA

## Architecture

Comparing architectures, the iPLD610 offers more flexibility than the 20RA10. This is based on macrocell count, total p-terms/macrocell, register type selections, and clock options. The iPLD610 offers 6 more macrocells than the 20RA10. It also has 8 dedicated sum-of-product p-terms in addition to asynchronous register reset (clear) and async clock/OE p-terms. The 20RA10 has a total of eight p-terms for each macrocell, only four of which are dedicated to performing standard sum-of-products logic. Register type selections offered by the iPLD610 include D, T, RS and JK, while the 20RA10 implements only a D-type register. As mentioned in the section on upgrading from the 22V10, register type selection can have a significant impact on the number of p-terms required to implement a logic function and may allow a design to "fit" with one type, but not with another. Also, of note is the fact that the iPLD610 can implement synchronous (via 2 clock inputs) or asynchronous (via individual macrocell p-terms) registered logic. The designer using the 20RA10 can only implement asynchronous registered logic (using clock p-term) only. As mentioned earlier

the synchronous register specifications of the iPLD610 ( $t_{SU}$ ,  $t_H$  and  $t_{CO1}$ ) are slightly skewed from the asynchronous register specifications ( $t_{ASU}$ ,  $t_{AH}$  and  $t_{ACO1}$ ). A designer using the iPLD610 can select clocking options on a macrocell-by-macrocell basis to best meet the needs of the system.

## D.C. Specifications

Looking at D.C. specifications of these devices, the 20RA10 and the iPLD610 exhibit similar maximum supply current. But the iPLD610 offers a programmable power-down mode, which can put this device into a power saving mode useful in many applications. The  $I_{OL}$  of the iPLD610 is higher than the 8 mA maximum for the 20RA10.

## 8.0 SUMMARY

The iPLD610 represents the highest performer in one of the industry-standard PLD architectures. The advantages gained by using the iPLD610 extend past the

obvious performance advantages of a 10 ns propagation delay and counter capability. The iPLD610 provides the following capabilities:

- 12 mA  $I_{OL}$
- Programmable Standby Mode
- 105 mA (max)  $I_{CC}$
- 16 Macrocell Architecture
- 100% Testability due to EPROM Technology
- 111 MHz Register Operation

In addition, this device is offered with several speed and package options. The device is available in 10 ns, 15 ns and 25 ns  $t_{PD}$  speeds. The packages for the iPLD610 are Plastic DIP and PLCC.

Already highlighted were the architectural advantages of this device over standard programmable logic devices. These advantages include device level capabilities (16 macrocells, up to 20 inputs, dual clocking) and macrocell level capabilities (total of 10 p-terms per macrocell, register type selection, feedback control, async clocking). These capabilities of the iPLD610

compared to competitive devices showed clearly this device outperforms others at a data sheet level (A.C. and D.C. specifications). But, the iPLD610 was further highlighted by the characterization data presented, which demonstrates performance outside data sheet considerations. The combination of data sheet and device characterization information provides designers an added level of confidence in actual device behavior, and knowledge of actual in-system performance.

The iPLD610 device is well supported by PLD design and development tools from major third party vendors as well as Intel's own PLDshell Plus software.

One other advantage of the iPLD610 is the ease of upgrade it provides to systems using the 5C060, EP610/630, 22V10 or 20RA10. These devices can be quickly upgraded to the iPLD610  $\mu$ PLD to gain performance, architectural, and power dissipation advantages.

The iPLD610 takes this standard architecture to a higher level. Its high performance in all areas open new applications to Intel  $\mu$ PLDs. Designers can now have both integration and performance!



November 1992

# iPLD22V10/85C22V10 Design Guide

**TODD WILSON, ROBERT JOHNSTON,  
AND LUAN NGUYEN**  
INTEL CORPORATION  
PROGRAMMABLE LOGIC APPLICATIONS

Order Number: 292106-001

# iPLD22V10/85C22V10 Design Guide

## CONTENTS PAGE

### 1.0 INTRODUCTION

Background .....	3-70
Document Overview .....	3-70

### 2.0 OVERVIEW

Product Overview .....	3-70
Features .....	3-71
Architecture .....	3-71

### 3.0 SPECIFICATION ANALYSIS

D.C. Characteristics .....	3-74
A.C. Characteristics .....	3-75

## CONTENTS PAGE

### 4.0 ADVANCED DESIGN ISSUES

Output Slew Rates .....	3-77
Ground Bounce .....	3-78
Propagation Delay	
versus number of p-terms .....	3-79
versus number of outputs	
switching .....	3-80
versus load capacitance .....	3-80
Clock to Valid Output .....	3-81
I/O Drive .....	3-81
I <sub>CC</sub> .....	3-82
Metastability .....	3-82

### 5.0 DESIGN EXAMPLES

Split Phase State Machine .....	3-83
Status Register .....	3-85

3

## 2.0 OVERVIEW

### Product Overview

The 22V10 architecture is one of several industry standard FLD architectures. While bipolar implementations of this architecture have worked well in the past, the speed and power consumption requirements of today's applications are beyond the limitations of bipolar devices. Intel's iPLD22V10 and 85C22V10 FLDs, based on CMOS LSI technology, meet the power and speed requirements of today's high-speed systems while maintaining pin- and I/O-compatible with standard 22V10 devices.

### Document Overview

This design guide provides technical support for design and design review, and offers interested in using in-

### Related Document Listing

Order Number	Document
280403	MOS Programmable Logic Device (PLD) Family Data Sheet
280416	22V10 Data Sheet
280417	Metastability Characteristics of Intel FLDs
280402	PLD Quality and Reliability Data Summary

## 1.0 INTRODUCTION

### Background

For years, designers have grown to depend on Programmable Logic Devices (PLDs) for low cost, high performance implementation of random logic and interface circuits. The combination of low cost, high performance, flexibility, and programming/development support make PLDs very attractive to both designers and design managers.

Intel's Programmable Logic Devices (PLDs) meet today's system requirements, as well as map the path to the needs of next generation's products. Intel PLDs provide the high-speed CMOS logic solution required by current and future microprocessors and VLSI peripherals.

The Intel PLD22V10 is a low cost, high-speed, low-power upgrade from PAL and GAL devices. Pin- and JEDEC-compatibility with industry standard 22V10s make the iPLD22V10 a drop-in upgrade requiring NO additional engineering effort.

The 85C22V10 PLD provides all the same features as the iPLD22V10. In addition, the 85C22V10's enhanced macrocell architecture gives the designer a greater range of feedback and clock options without sacrificing pin- and JEDEC-compatibility with industry standard 22V10 parts. The 85C22V10 may be programmed as a standard 22V10 device by using a standard 22V10 JEDEC file. Optional *superset* features are accessed through an extended JEDEC file compiled by popular design tools such as ABEL and CUPL, or through Intel's PLDshell Plus.

### Document Overview

This design guide provides technical support for designers, design managers, and others interested in using Intel's

PLD22V10 and 85C22V10 PLDs. The information contained in this document is intended to support both the decision making process prior to design and the qualification process that occurs during and after the design is complete. The format of the design guide is as follows:

**Section 2-Product Overview:** Highlights and architecture of the iPLD22V10 and the 85C22V10 PLDs.

**Section 3-Specification Analysis:** The key D.C. and A.C. specs (from the data sheet) are discussed and compared against competitive devices. This section provides a baseline for comparison and device selection. Also, some insights are provided on how to best use the data sheet specifications.

**Section 4-Advanced Design Issues:** This section discusses issues affecting high-speed systems designs. Topics include output slew rates, effects of capacitive loading on outputs, and synchronous/asynchronous register operation.

**Section 5-Design Examples:** Design ideas using the unique performance and architecture combination of the Intel 85C22V10 are presented.

## 2.0 OVERVIEW

### Product Overview

The 22V10 architecture is one of several industry standard PLD architectures. While bipolar implementations of this architecture have worked well in the past, the speed and power consumption requirements of today's applications are beyond the limitations of bipolar devices. Intel's iPLD22V10 and 85C22V10 PLDs, based on CHMOS IIIIE technology, meet the power and speed requirements of today's high-speed systems while maintaining pin- and JEDEC-compatibility with standard 22V10 devices.

Related Document Listing	
Document	Order Number
CMOS Programmable Logic Device (PLD) Family Data Sheet (Includes iPLD22V10, iPLD610, and iPLD910)	290453
85C22V10 Data Sheet	290416
Metastability Characteristics of Intel EPLD's	292071
PLD Quality and Reliability Data Summary	293003

## Features

The iPLD22V10 and 85C22V10 are high-performance, high-integration, general-purpose CMOS PLDs. The features provided by these devices include:

- High Speed Operation ( $t_{PD} = 10$  ns,  $t_{C01} = 7$  ns, 71.4 MHz State Machine Frequency, 100 MHz with No Feedback)
- Typical  $I_{CC} = 90$  mA @ 15 MHz
- 10 Programmable Macrocells (I/O Pins)
- EPROM Cell, CMOS Technology
- 100% Silicon Testability
- 24-pin DIP, 28-pin PLCC Packages
- Programmable Security Bit
- *Superset* Clock and Feedback Features (85C22V10)

## PACKAGING

Figure 1 shows the pinouts of the DIP and PLCC packages for the iPLD22V10 and 85C22V10 PLDs. Both the iPLD22V10 and the 85C22V10 are available in a plastic One-Time-Programmable (OTP) DIP package and in a plastic OTP PLCC package. In addition, the 85C22V10 is available in a UV erasable/reprogrammable Ceramic DIP package.

## PROCESS

The high performance of the iPLD22V10 and 85C22V10 is a result of combining an industry-standard

architecture with Intel's advanced 1-micron CHMOS III-E EPROM technology. This technology brings high speed and significant power savings to a well known architecture.

## TESTABILITY

The CHMOS III-E EPROM technology used in the iPLD22V10 and 85C22V10 allows complete testability of every part Intel produces. At the wafer level, all device programmable elements are programmed and tested, and all data paths are 100% tested. The device is then completely erased. Once the device is packaged, special manufacturing test modes are utilized to ensure post packaging device functionality and performance. Intel's rigorous test procedures ensure high yields for our customers, and high reliability for their customers.

## Architecture

### GLOBAL ARCHITECTURE

Figure 2 shows the global architecture for both devices. The iPLD22V10 and 85C22V10 feature 12 dedicated input pins (including CLK/INP0) and 10 I/O pins. Each of the I/O pins is associated with a macrocell. All inputs and feedback signals and their compliments are available to all product (AND) terms. The product terms are summed by a single OR-gate, thus forming an AND-OR logic array. The sum-of-products (SOP) is fed into the macrocell. Each macrocell is fed by a maximum of from 8 to 16 product terms, depending on the position of the particular macrocell.

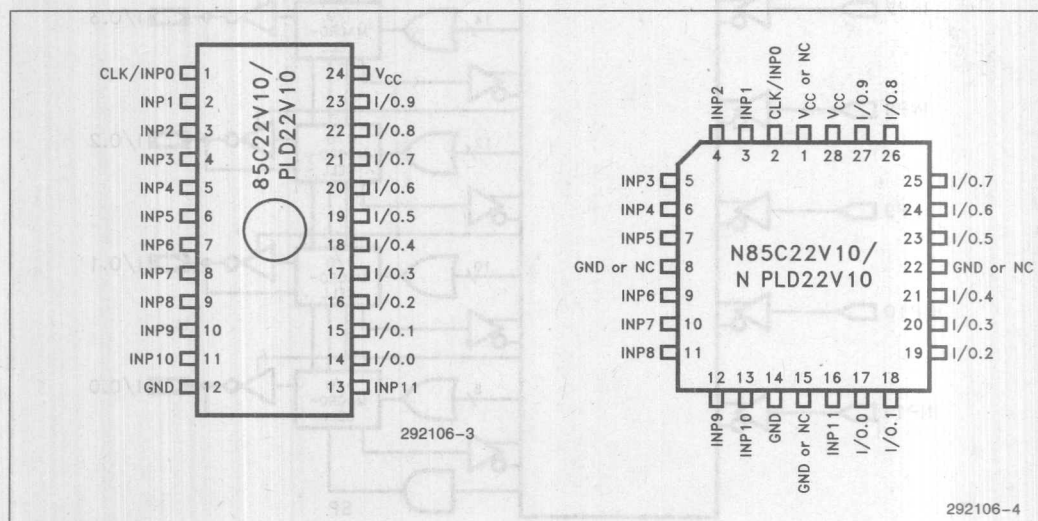


Figure 1. Pinout Diagrams for iPLD22V10 and 85C22V10



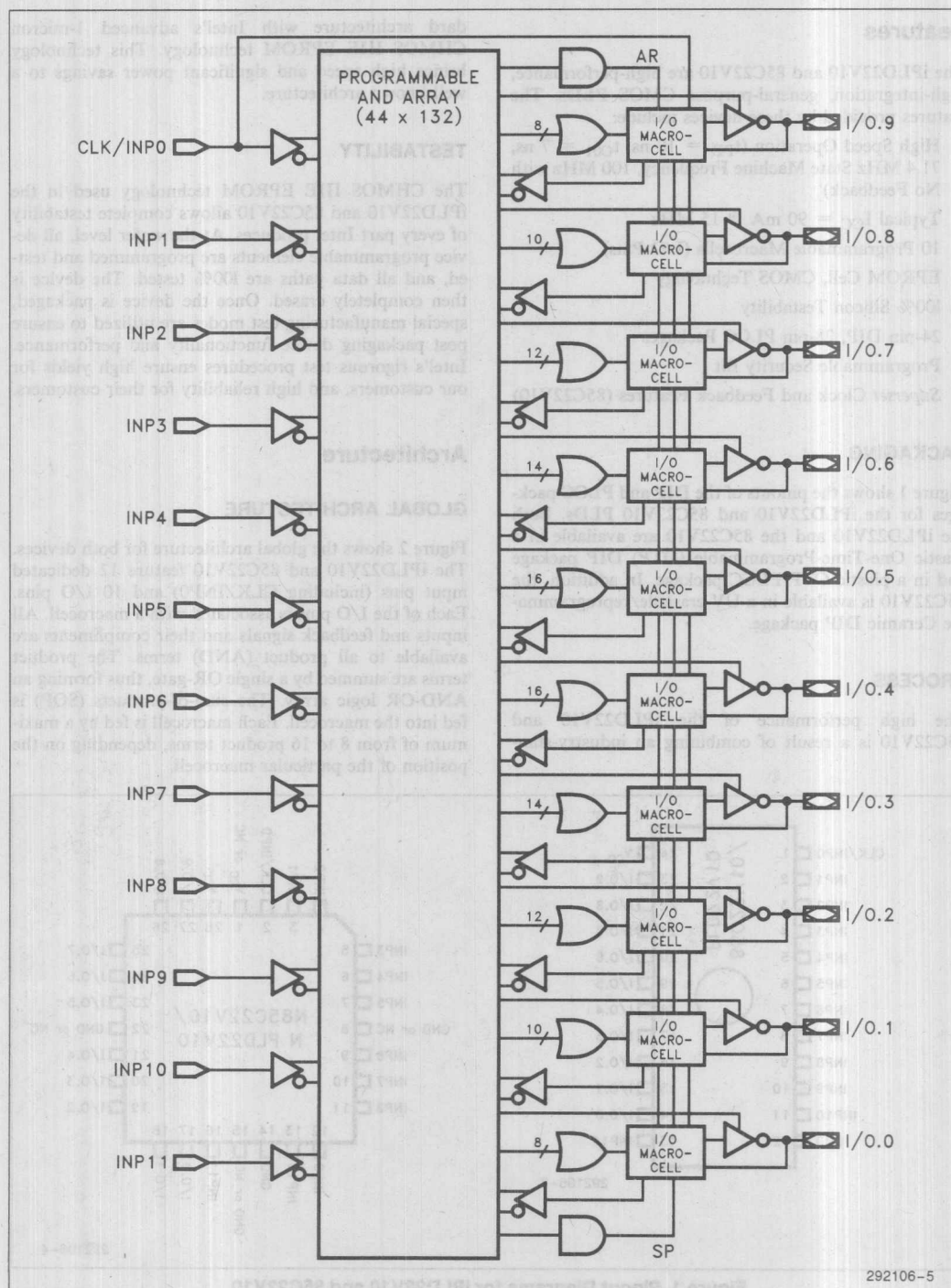


Figure 2. iPLD22V10/85C22V10 Global Architecture

## GLOBAL PRESET AND RESET

The iPLD22V10 and 85C22V10 registers may be synchronously set and asynchronously reset on a global basis. The synchronous preset signal (SP) is determined for all macrocells by a single global p-term that is separate from macrocell sum-of-products (SOP). After SP is asserted, the next clock transition that triggers the macrocell registers will cause the Q outputs to be set to logic HI. Note that if post-register inversion is programmed (active-low output), the macrocell's pin will be set to a logic LOW.

The asynchronous reset signal (AR) is also derived from a single independent global p-term. Macrocell registers are reset by AR asynchronously with respect to the device clock. If post-register inversion is programmed (active-low output), the macrocell's pin will be reset to a logic HI. For details on timing parameters that relate to preset and reset, please refer to A.C. Specifications.

## Macrocell Architecture

### iPLD22V10

Figure 3 shows the macrocell architecture for the iPLD22V10. The architecture allows registered or combinatorial logic with active-high or active-low output. Note that in the iPLD22V10, the output type determines the feedback type (if any feedback is used): registered output dictates registered feedback, and combinatorial output dictates pin feedback. Also, because outputs may be programmed as either active-high or active-low on a macrocell-by-macrocell basis, the polarity or "sense" of the output pin associated with a macrocell register is independent of the register itself. Table 1 lists the output configurations for the iPLD22V10.

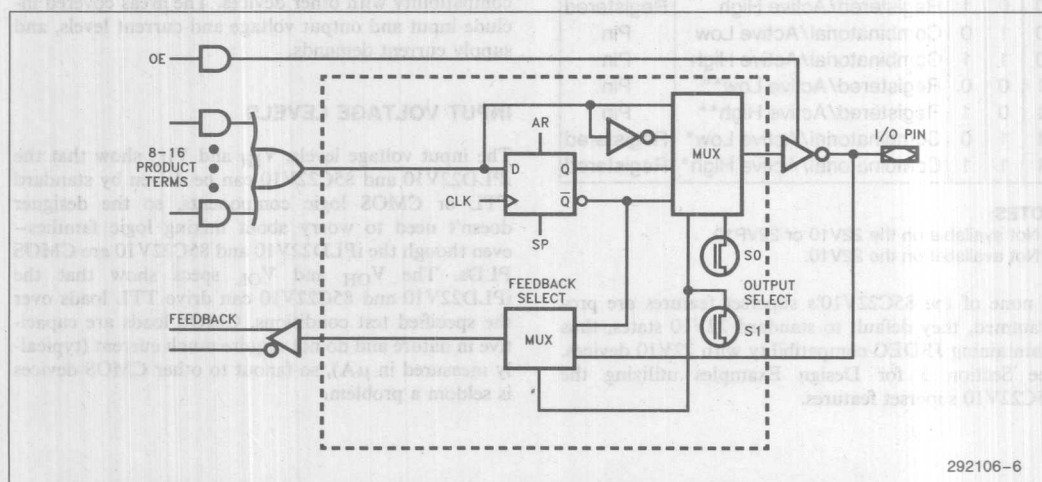


Figure 3. iPLD22V10 Macrocell Architecture

3-73

Table 1. iPLD22V10 Macrocell Configurations

S1	S0	Output/Polarity	Feedback
0	0	Registered/Active Low	Registered
0	1	Registered/Active High	Registered
1	0	Combinatorial/Active Low	Pin
1	1	Combinatorial/Active High	Pin

The Clock (CLK) input for macrocell registers comes from PIN 1. If a CLK signal is not needed, i.e. if all outputs are combinatorial, PIN 1 may be used as another generic input to the logic array. Output Enable (OE) for each macrocell is controlled by a single p-term in the logic array. This p-term is not taken from the product terms that feed the macrocell (as is the case in traditional PAL devices), so there is no penalty for using the OE p-term. OE for each macrocell is independent and asynchronous.

### 85C22V10

The 85C22V10 macrocell architecture is shown in Figure 4. This is an enhanced version of the iPLD22V10 macrocell. Note the additions to the clock and feedback selection sections. The addition of the XOR gate at the clock input to the macrocell flip-flop allows the clock to be inverted. When this feature is programmed, the macrocell flip-flop will latch at the falling edge of the CLK input, rather than the rising edge. This gives the system designer greater flexibility to determine clock timing for each macrocell.

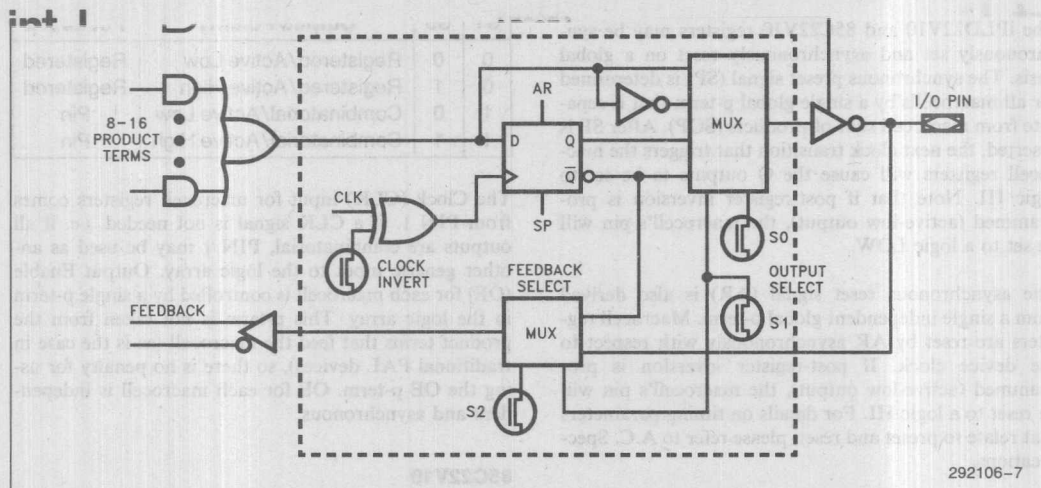


Figure 4. 85C22V10 Macrocell Architecture

The other addition to this macrocell is the S2 switch in the feedback selection. The effect of S2 is clearly seen in Table 2. All combinations of output and feedback can now be supported. This opens up the possibility of using the macrocell pins as true I/O ports. For instance, a bi-directional register can now be implemented on a single 85C22V10. The 85C22V10 also allows a combinatorial signal to drive a macrocell pin while a registered version of that signal is fed back into the device. See Design Example 2.

Table 2. 85C22V10 Macrocell Configurations

S2	S1	S0	Output/Polarity	Feedback
0	0	0	Registered/Active Low	Registered
0	0	1	Registered/Active High	Registered
0	1	0	Combinatorial/Active Low	Pin
0	1	1	Combinatorial/Active High	Pin
1	0	0	Registered/Active Low**	Pin
1	0	1	Registered/Active High**	Pin
1	1	0	Combinatorial/Active Low*	Registered
1	1	1	Combinatorial/Active High*	Registered

**NOTES:**

\*Not available on the 22V10 or 22VP10.

\*\*Not available on the 22V10.

If none of the 85C22V10's superset features are programmed, they default to standard 22V10 states, thus maintaining JEDEC compatibility with 22V10 devices. See Section 5 for Design Examples utilizing the 85C22V10 superset features.

### 3.0 SPECIFICATION ANALYSIS

This section is intended to expand on some of the data sheet details. Those qualities that the specifications characterize are interpreted.

Typical data taken from in-circuit applications is presented in Section 4, Advanced Design Issues.

### D.C. Characteristics

D.C. characteristics describe the steady-state behavior of a device. These specs provide insight into the iPLD22V10/85C22V10's power consumption and compatibility with other devices. The areas covered include input and output voltage and current levels, and supply current demands.

### INPUT VOLTAGE LEVELS

The input voltage levels,  $V_{IH}$  and  $V_{IL}$ , show that the iPLD22V10 and 85C22V10 can be driven by standard TTL or CMOS logic components, so the designer doesn't need to worry about mixing logic families—even though the iPLD22V10 and 85C22V10 are CMOS PLDs. The  $V_{OH}$  and  $V_{OL}$  specs show that the iPLD22V10 and 85C22V10 can drive TTL loads over the specified test conditions. CMOS loads are capacitive in nature and do not require much current (typically measured in  $\mu A$ ), so fanout to other CMOS devices is seldom a problem.

## OUTPUT SPECIFICATIONS: $I_O$ AND $V_O$

The best measure of a device's ability to handle loads is  $I_{OL}$ . Most devices can "sink" more current than they can "source", so  $I_{OL}$  is usually quoted in connection with load driving ability. The data sheet test condition for  $V_{OL}$  shows that the iPLD22V10 and 85C22V10 are guaranteed to handle 16 mA loads while maintaining the output voltage at or below 0.45V.

$I_{OH}$  is generally specified at minimum voltage-high output,  $V_{OH}$ . The iPLD22V10 and 85C22V10 outputs are guaranteed to supply -4 mA at a minimum of 2.4V. This is information applicable for driving TTL loads, but CMOS devices will consume less current internally if their inputs are driven closer to  $V_{CC}$ . For this reason, Intel PLD22V10 and 85C22V10 PLDs have a built-in pullup circuit to drive their outputs to near  $V_{CC}$ , and are specified to provide a minimum of -100  $\mu$ A for  $V_O = V_{CC} - 0.3V$ . Refer to Figure 14 for typical  $I_O$  vs  $V_O$  data.

## $I_{CC}$ SPECIFICATIONS

The D.C. specification of greatest interest to most designers is  $I_{CC}$ . This value not only specifies how much current the device will require, but also indicates how much heat (watts) the device will dissipate. For this reason, the  $I_{CC}$  specification will affect both the power supply requirements and the board reliability.  $I_{CC}$  is measured with the device unloaded, so the value indicates how much current the device itself requires.

$I_{CC}$  is specified at 15 MHz, but because CMOS devices consume most of their power during transitions,  $I_{CC}$  varies with the frequency of the clock and other inputs. For more precise current/power calculations see Figure 5: Typical  $I_{CC}$  vs Frequency for a 10-bit counter.

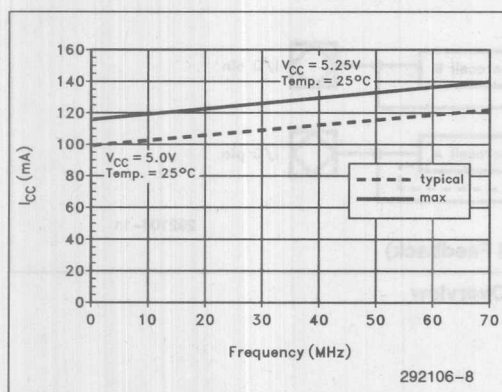


Figure 5. iPLD22V10/85C22V10-10  $I_{CC}$  vs Frequency

## A.C. Characteristics

The A.C. specifications describe the guaranteed switching frequencies and timing properties of the device. In this section, the meanings of data sheet quantities are clarified, and comparisons are made with competitive parts.

### COMBINATORIAL PARAMETERS

Timing for combinatorial functions is determined by the total pin-to-pin propagation delay and the output enable/disable times  $t_{PD}$ ,  $t_{PZX}$  and  $t_{PXZ}$ .  $t_{PD}$  is the delay between the time that valid data is present on an input pin, and the time that the effect of that input data is seen at an output pin.  $t_{PD}$  is measured with output enable asserted; it does not include  $t_{PZX}$  or  $t_{PXZ}$ . Figure 6 shows  $t_{PD}$ ,  $t_{PZX}$  and  $t_{PXZ}$  in the form of a timing diagram.

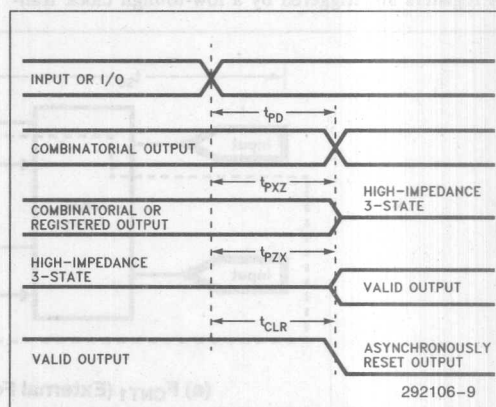


Figure 6. Combinatorial Timing Parameters

### SYNCHRONOUS PARAMETERS

Registered output requires that valid data be present at the register's data input before the register is clocked. The user of a device ensures that this condition is met by adhering to the setup time,  $t_{SU}$ . The setup time determines how long before the clock edge arrives at the CLK pin that valid data must already be at the input pin(s).

The time between the arrival of the clock edge at the CLK pin and valid data being present at the output pin is  $t_{CO1}$ . The minimum time needed to affect a valid registered output is then  $t_{SU} + t_{CO1}$ . If the device is configured as a counter using external feedback from a macrocell output pin to an input pin, the maximum frequency with which it can be clocked is  $F_{CNT1} = 1/(t_{SU} + t_{CO1})$ .



If internal feedback is used, then input and output buffers are no longer in the data path. Propagation time is decreased, and the maximum counter frequency is increased to  $F_{CNT2}$ . Figure 7 provides an overview of both  $F_{CNT}$  values.

The device may be used to output a registered sum of products without feedback. In this case, the maximum clock frequency is limited by the minimum clock period:  $f_{MAX} = 1/t_{CP}$ .

### PRESET AND RESET SPECIFICATIONS

The **Synchronous Preset (SP)** signal is derived from a single independent p-term. To insure predictable preset results, the designer will make sure that SP will meet the Synchronous Preset to CLK  $\uparrow$  Setup Time,  $t_{SP}$ , before a clock edge is applied to the CLK pin to trigger a global register preset. In the case of the iPLD22V10, all registers are triggered by a low-to-high clock tran-

sition, and  $t_{SP}$  is measured from that transition. However, in the case of the 85C22V10, the sense of the CLK signal can be selected for each macrocell on an individual basis. The astute designer will see that this raises the possibility of synchronously presetting a subset of the registers on the device.

For example, consider two state machines implemented on the same 85C22V10. The first state machine's registers are programmed to respond to a low-to-high clock transition at the CLK pin, while the second state machine's registers are programmed to respond to the inverted high-to-low clock transition. If SP is asserted  $t_{SP}$  before a low-to-high clock transition, then the first state machine will be preset. But if SP is unasserted  $t_{SP}$  before the high-to-low clock transition, the second state machine will NOT be preset. In a similar fashion, the second state machine may be preset without affecting the first. A timing diagram of this *superset* capability is presented in Figure 8.

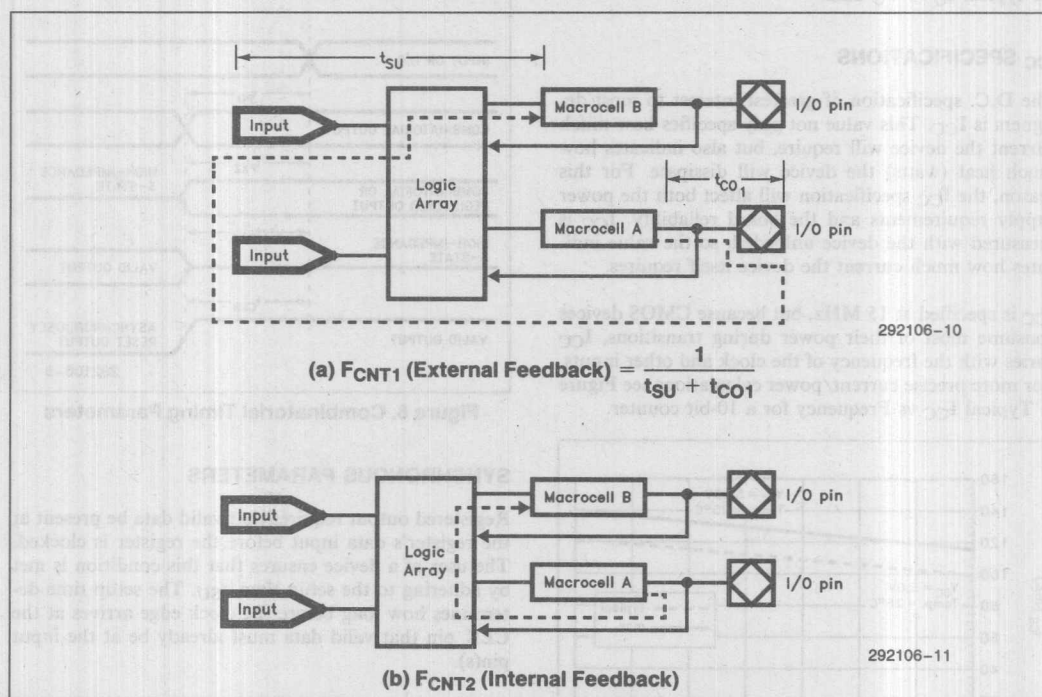


Figure 7.  $F_{CNT}$  Overview

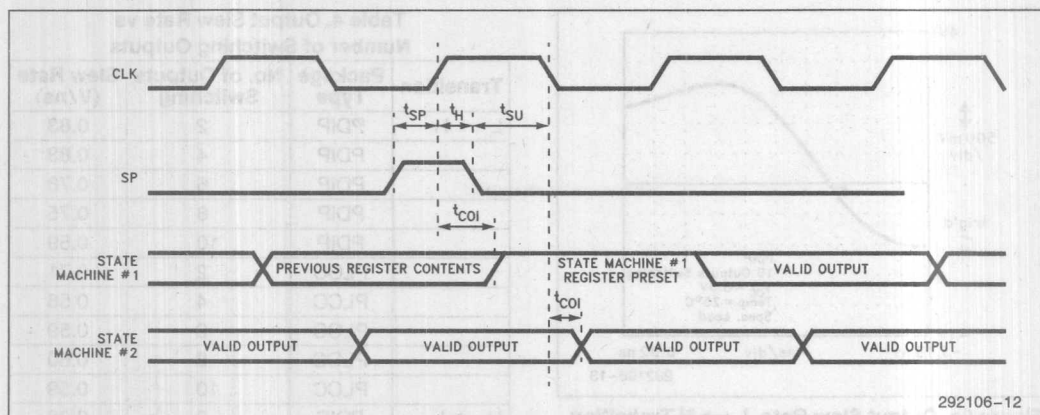


Figure 8. Dual State Machine Synchronous Preset

The **Asynchronous Reset (AR)** signal is also determined by a single global p-term. However, because this is an asynchronous signal, no setup time with respect to CLK  $\uparrow$  is required. The time that elapses between the assertion of the reset condition and the time that the actual reset occurs at the output pin is  $t_{CLR}$ .

**Asynchronous Reset Recovery time,  $t_{ARR}$** , is the *maximum* amount of time required for macrocell registers to begin correctly responding to CLK after AR is removed. No minimum recovery time is specified. To insure predictable results, AR should be removed (unasserted) at least  $t_{ARR}$  before the next active clock edge appears at the CLK input. Other approaches here might be to disable outputs or hold inputs constant until  $t_{ARR}$  after AR is removed. Since  $t_{ARR}$  is a setup time with respect to CLK, it is given as a synchronous specification. On the other hand,  $t_{CLR}$  does not depend on CLK, so it is included in the combinatorial specifications.

#### 4.0 ADVANCED DESIGN ISSUES

As end users demand higher speeds and time to market becomes more critical, designers require more detailed information than is typically available in the data sheet. The purpose of this section is to provide designers with data that characterizes the effects of loading, temperature, and internal resource utilization on device performance. This data is intended to help support engineering decisions in all phases of product development, from design to production.

Data were measured with the output load specified in the 85C22V10 data sheet unless otherwise specified. The topics covered in this section are:

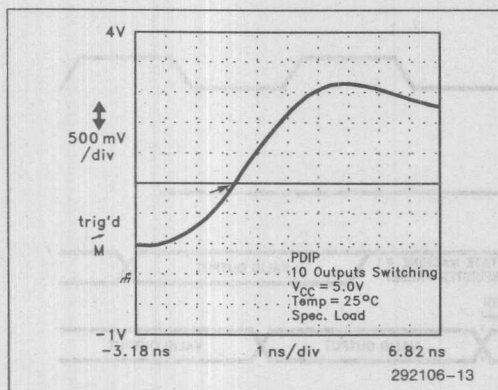
- Output Slew Rates
- $t_{PD}$  Characteristics for Combinational Logic
- Output Skew Characteristics for Synchronous Register Operation
- Asynchronous Register Operation Characteristics
- Output Current Characteristics
- Design Considerations

#### Output Slew Rate (Edge Rate)

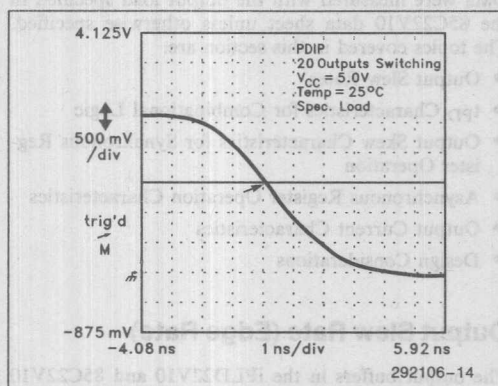
The output buffers in the iPLD22V10 and 85C22V10 are designed to reduce transmission line effects through controlling output slew rates. Rapid voltage changes contain high-frequency components that can produce unwanted transmission line effects on circuit board traces. Possible effects include ringing, dispersion, and radiation losses. Overcoming these effects can be time consuming and expensive. The iPLD22V10 and 85C22V10 output buffer design limits transmission line effects through providing low output slew rates. These low slew rates contain fewer of the high-frequency signal components that give board designers headaches.

Figure 9 provides sample iPLD22V10/85C22V10 output waveforms expanded to highlight slew rate. Tables 3 and 4 show that the slew rate is very consistent with respect to both temperature and number of outputs switching, though the slew rate does decrease slightly as temperature or the number of switching outputs increases.

The consistent, slow edge rates provided by the Intel PLD22V10 and 85C22V10 allow the designer to use the full capabilities of these devices without having to worry about transmission line effects.



**Figure 9a. Output Slew Rate, L → H Transition**  
(PDIP, 10 Outputs Switching,  $V_{CC} = 5V$ ,  
Temp = 25°C, Spec. Load)



**Figure 9b. Output Slew Rate, H → L Transition**  
(PDIP, 10 Outputs Switching,  $V_{CC} = 5V$ ,  
Temp = 25°C, Spec. Load)

**Table 3. Output Slew Rate vs Temperature**

Transition	Package Type	Temperature (°C)	Slew Rate (V/ns)
L → H	PDIP	0	0.72
	PDIP	25	0.72
	PDIP	70	0.70
	PLCC	0	0.56
	PLCC	25	0.59
H → L	PLCC	70	0.53
	PDIP	0	0.67
	PDIP	25	0.67
	PDIP	70	0.62
	PLCC	0	0.87
	PLCC	25	0.80
	PLCC	70	0.77

**TEST CONDITIONS:**

$V_{CC} = 5V$ , 10 Outputs Switching, Spec. Load.

**Table 4. Output Slew Rate vs  
Number of Switching Outputs**

Transition	Package Type	No. of Outputs Switching	Slew Rate (V/ns)
L → H	PDIP	2	0.83
	PDIP	4	0.83
	PDIP	6	0.78
	PDIP	8	0.75
	PDIP	10	0.59
	PLCC	2	0.71
	PLCC	4	0.56
	PLCC	6	0.59
	PLCC	8	0.60
	PLCC	10	0.59
H → L	PDIP	2	0.88
	PDIP	4	0.78
	PDIP	6	0.75
	PDIP	8	0.70
	PDIP	10	0.67
	PLCC	2	1.17
	PLCC	4	1.0
	PLCC	6	1.0
	PLCC	8	1.0
	PLCC	10	0.80

**TEST CONDITIONS:**

$V_{CC} = 5V$ , Temp = 25°C, Spec. Load.

## Ground Bounce

The rapid current changes that occur inside all high-speed logic devices during normal state transitions cause short-term voltage changes, or "glitches", on pins that are not changing state. This phenomenon is called *ground bounce*. Ground bounce is typically seen during the rising or falling edges of a state change, where internal currents must change most rapidly.

Table 5 shows typical ground bounce levels for the Intel iPLD22V10 and 85C22V10 PLDs compared to several competing parts. Data presented in Table 5 was taken using the load specified in "EDN's advanced CMOS logic ground-bounce tests", EDN, March 2, 1989.

The low ground bounce levels of the Intel PLDs guard against glitch problems in high-performance systems, and free high-performance designers from a major headache.

The Intel iPLD22V10 and 85C22V10 PLDs in the plastic PLCC package provide the option of additional ground and  $V_{CC}$  connections. Table 6 shows the effect of using these additional connections. While the use of these pins is not required for normal specified operation, connecting these pins produces a positive effect on ground bounce and glitch immunity.

Table 5. Ground Bounce vs Competition

	Intel 85C22V10	AMD PALCE22V10	Lattice GAL22V10	Cypress PAL22V10	ICT PEEL22V10
Package	PDIP	PDIP	PDIP	PDIP	PDIP
Pk to Pk	1.40V	1.48V	1.84V	1.40V	1.56V
Peak	0.800V	0.92V	1.04V	0.76V	0.92V

**TEST CONDITIONS:**V<sub>CC</sub> = 5V, Temperature = 25°C, EDN Load.Table 6. Ground Bounce with Additional V<sub>CC</sub> and GND Pins

	Intel 85C22V10	Intel 85C22V10
Package Connections	PLCC Pins 1, 8, 15, 22 Open (NC)	PLCC Pins 1, 8, 15, 22 Connected
Pk to Pk	1.48V	1.32V
Peak	0.88V	0.68V

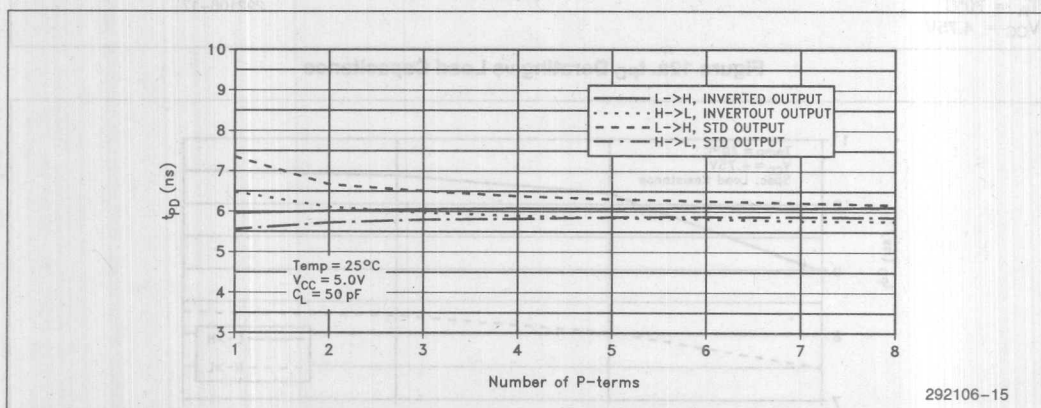
**TEST CONDITIONS:**V<sub>CC</sub> = 5V, Temperature = 25°C, Spec Load.**Propagation Delay: t<sub>PD</sub>**

The maximum propagation delay (t<sub>PD</sub>) specified in the data sheet for -10 devices is 10 ns. This is guaranteed for worst-case voltage and temperature while driving the spec load. Factors that effect typical t<sub>PD</sub> include the number p-terms in the data path, the number of outputs switching simultaneously, and load capacitance.

Figure 10 shows typical t<sub>PD</sub> vs p-terms characteristics. As the number of p-terms increases, internal device

loading causes the propagation delay to increase or decrease slightly.

Figure 11 shows the relation of t<sub>PD</sub> to number of outputs switching. As the number of outputs simultaneously switching increases, t<sub>PD</sub> also increases. This effect is related to the ability of the package power and ground leads to channel additional current to the device.

Figure 10. t<sub>PD</sub> vs Number of P-Terms



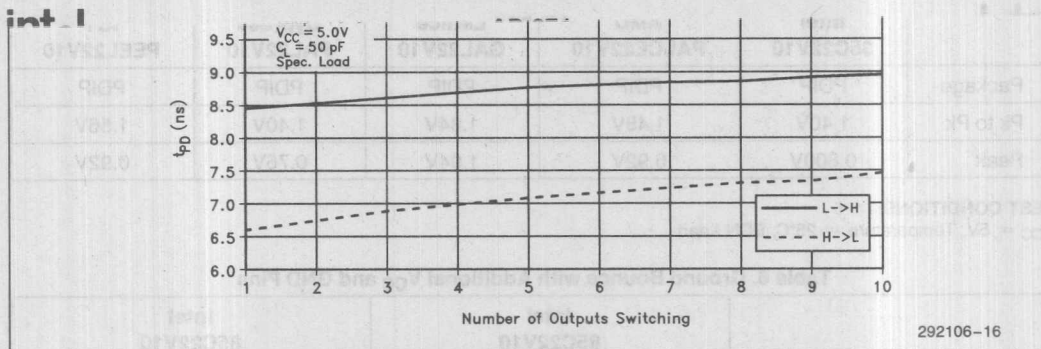


Figure 11.  $t_{PD}$  vs Number of Switching Outputs

Figure 12a is the data sheet derating curve for  $t_{PD}$  as a function of load capacitance. A large load capacitance will require a long time to charge up to an acceptable voltage level, and thus increase  $t_{PD}$ . Typical values for  $t_{PD}$  taken over a range of load capacitances are shown

in Figure 12b. The typical value for  $t_{PD}$  at 160 pF load capacitance is approximately 10.5 ns. Since  $t_{PD}$  is specified to be 10 ns max with a load capacitance of 50 pF, this represents a derating of 0.5 ns. This is well within the 3.5 ns maximum derating specified in Figure 12a.

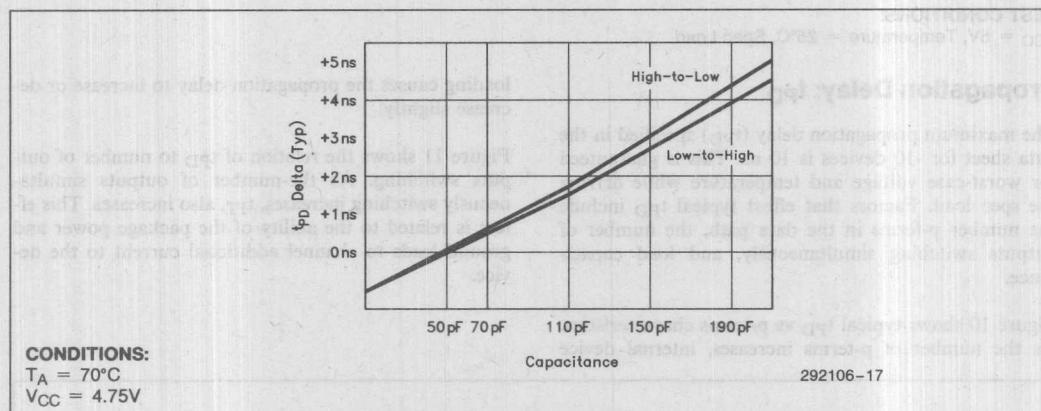


Figure 12a.  $t_{PD}$  Derating vs Load Capacitance

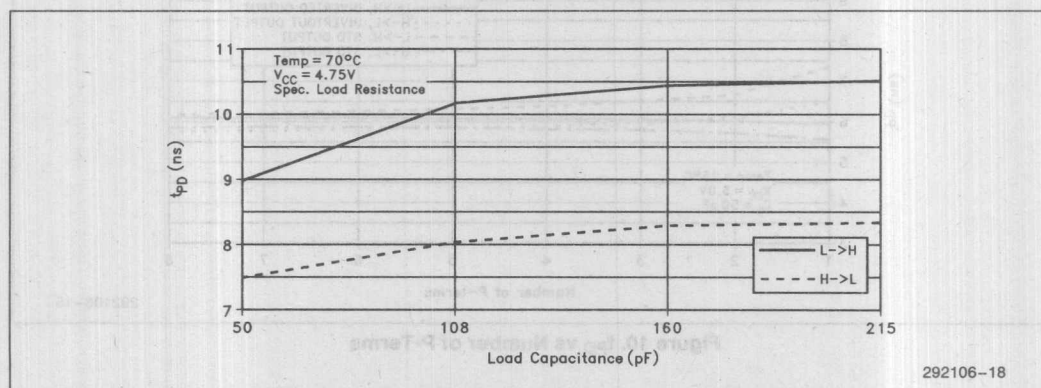


Figure 12b. Typical  $t_{PD}$  vs Load Capacitance

## CLK to Valid Output: $t_{CO1}$

The time that elapses between the appearance of an active clock transition at the CLK pin and the effect of that transition on data at an output pin is  $t_{CO1}$ . Intel PLDs are compensated to reduce the effect of temperature on performance. A  $t_{CO1}$  Derating vs Temperature curve is included in the datasheet.

$t_{CO1}$  skew is the difference in timing of one output pin with respect to another. This phenomenon is a result of different internal path lengths from the CLK pin to the various macrocells. Note that only minimum and maximum values are specified for  $t_{CO1}$ . No values for skew are quoted in the datasheet. Figure 13 shows typical  $t_{CO1}$  skew measurements for different temperatures. The greatest measured skew was well under 1 ns for L  $\rightarrow$  H transitions. The largest measured variation with respect to temperature was under 0.6 ns in the case of H  $\rightarrow$  L transitions.

A look at the 85C22V10 datasheet shows that the specified  $t_{CO1}$  ranges are identical for standard CLK and

optional inverted CLK. The architecture of the 85C22V10 macrocell reveals that the CLK signal passes through the XOR clock inversion gate whether or not clock inversion is used. Thus, programming the inverted clock *superset* feature introduces no speed penalty.

## I/O Drive

The ability of a device to drive a load is directly reflected by  $I_O$ . Obviously, output current depends on the nature of the load, so only the short-circuit current,  $I_{SC}$ , appears in the data sheet. Figure 14 gives greater insight into the output current capabilities of the iPLD22V10 and 85C22V10. This graph shows output current as a function of load voltage for both output-high and output-low conditions. Notice that in this graph, the polarity of the current is ignored. For the output-low condition ( $I_{OL}$ ), positive conventional current is flowing into the device: the device is sinking current. For output-high condition ( $I_{OH}$ ), current is flowing out of the device: the device is sourcing current.

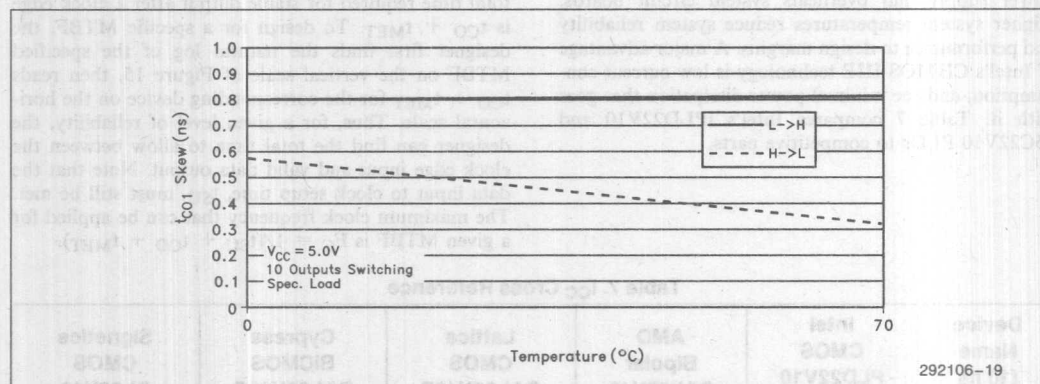
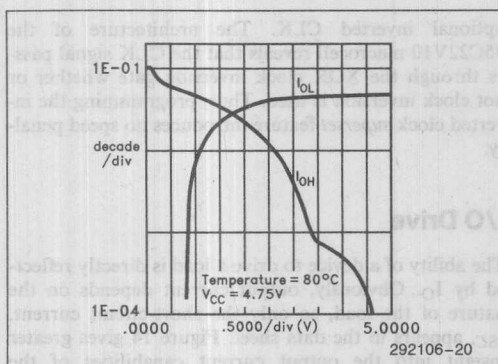


Figure 13. Typical  $t_{CO1}$  Skew vs Temperature

Figure 14.  $I_O$  vs  $V_O$ 

### Current Consumption: $I_{CC}$

The best indication of a device's power consumption is  $I_{CC}$ . One of the limitations of older bipolar devices is their high current consumption that requires a large power supply and overheats system circuit boards. Higher system temperatures reduce system reliability and performance to design margins. A major advantage of Intel's CHMOS IIIIE technology is low current consumption, and the minimal power dissipation that goes with it. Table 7 compares Intel's iPLD22V10 and 85C22V10 PLDs to competitive parts.

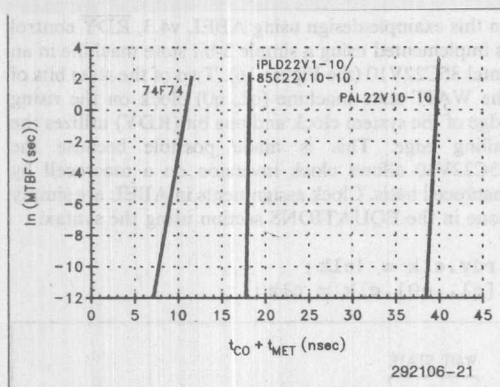
### Metastability

"Metastable" may be defined as the condition of a bi-stable system before it has entered one of its stable states. Metastability may be perceived as oscillation or "hovering" between stable states for a short period of time before the system resolves into a stable state. Metastability may be introduced into a logic system by violating timing or voltage parameters laid out in data sheets, but it may also appear as a result of random noise, ambient conditions, or other conditions beyond the control of the system designer. For this reason, metastability is usually predicted on a statistical basis and measured as Mean Time Between Failures (MTBF) for a given variable.

Figure 15 shows typical MTBF as a function of  $t_{CO} + t_{MET}$  for the iPLD22V10/85C22V10-10 and for the PAL 22V10-10. These parts were programmed as 10-bit counters. Data for the 74F74 D flip-flop is also presented for comparison.  $t_{CO}$  is taken from the data sheet.  $t_{MET}$  is the amount of time after  $t_{CO}$  that is required for the device to produce a stable output. The total time required for stable output after a clock edge is  $t_{CO} + t_{MET}$ . To design for a specific MTBF, the designer first finds the natural log of the specified MTBF on the vertical scale of Figure 15, then reads  $t_{CO} + t_{MET}$  for the corresponding device on the horizontal scale. Thus, for a given level of reliability, the designer can find the total time to allow between the clock edge input and valid data output. Note that the data input to clock setup time,  $t_{SU}$ , must still be met. The maximum clock frequency that can be applied for a given MTBF is  $F_C = 1/(t_{SU} + t_{CO} + t_{MET})$ .

Table 7.  $I_{CC}$  Cross Reference

Device Name (10 ns Parts)	Intel CMOS PLD22V10 85C22V10	AMD Bipolar PAL22V10	Lattice CMOS GAL22V10B	Cypress BiCMOS PAL22V10C	Signetics CMOS PL22V10
$I_{CC}$ @ 15 MHz	130 mA	180 mA	130 mA	190 mA	120 mA + 0.5 mA/MHz



**Figure 15. Tau Curves for Intel PLD22V10/85C22V10-10 vs Competition**

The Intel PLD22V10 and 85C22V10 are protected from metastability through careful register design and the use of high-gain output amplifiers driven by the macrocell. This provides immunity from metastability by ensuring that the output will be driven to one

extreme state or the other: either logic HI or LOW. For more information on metastability, please refer to the Intel Application Note "Metastability Characteristics of Intel EPLDs", order number 292071-002.

## 5.0 DESIGN EXAMPLES

### Design Example # 1—Split Phase State Machine

#### INTRODUCTION

It is sometimes desirable in synchronous system design to have events occur on both the rising and falling edge of the system clock. This allows events to be processed with greater timing granularity.

This example is intended to illustrate one possible use of the programmable clock invert feature of the Intel 85C22V10. In doing this, it is also shown how to access this feature using Data I/O's ABEL design tool.

3

```

module WAIT
title 'wait state controller'
wait device 'p22v10ic';      "Intel 85C22V10 in PLCC package
clk, cs0, cs1, cs2, cs3 pin 2, 3, 4, 5, 6;    "Inputs
rdy , q0, q1 pin 23, 24, 25;                "Outputs
"WAIT statemachine bits defined
start = ^h0; wait2 = ^h4; half1 = ^h2;
idle = ^h7; wait1 = ^h6;
wait3 = ^h5; ready = ^h3;
state_diagram [rdy, q1, q0]
state start: goto idle;
state idle: if ( !cs0 & cs1 & cs2 & cs3) then ready
            else if ( cs0 & !cs1 & cs2 & cs3 ) then wait1
            else if ( cs0 & cs1 & !cs2 & cs3 ) then wait2
            else if ( cs0 & cs1 & cs2 & !cs3 ) then wait3
            else idle;
state wait3: goto wait2;      "insert 3 waitstates
state wait2: goto wait1;     "insert 2 waitstates
state wait1: goto ready;     "insert 1 waitstate
state half1: goto ready;     "half state between wait1 and ready
state ready: goto idle;
equations
rdy.clk = !clk;              "specify /clk for rdy state bit
[q1..q0].clk = clk;
end WAIT

```

**Figure 16. ABEL Source Code for Split Phase State Machine**



## int-1

When the Intel 486 microprocessor communicates with memory or I/O devices, several mechanisms exist for extending the bus cycles to accommodate slow memory or I/O devices. Bus cycle extensions are usually referred to as wait-states. One method for requesting wait-states from the Intel 486 microprocessor during a bus cycle is to keep the  $\overline{\text{RDY}}$  line inactive until the desired number of wait states have been incurred.

Intel 85C22V10 (see Figure 16). Two of the state bits of the WAIT state machine (q1, q0) clock on the rising edge of the system clock and one bit (RDY) utilizes the falling edge. This is made possible because the 85C22V10 allows clock inversion on a macrocell by macrocell basis. Clock assignments in ABEL are simply done in the EQUATIONS section using the syntax:

```
rdy.clk = !clk;
[q1..q0].clk = clk;
```

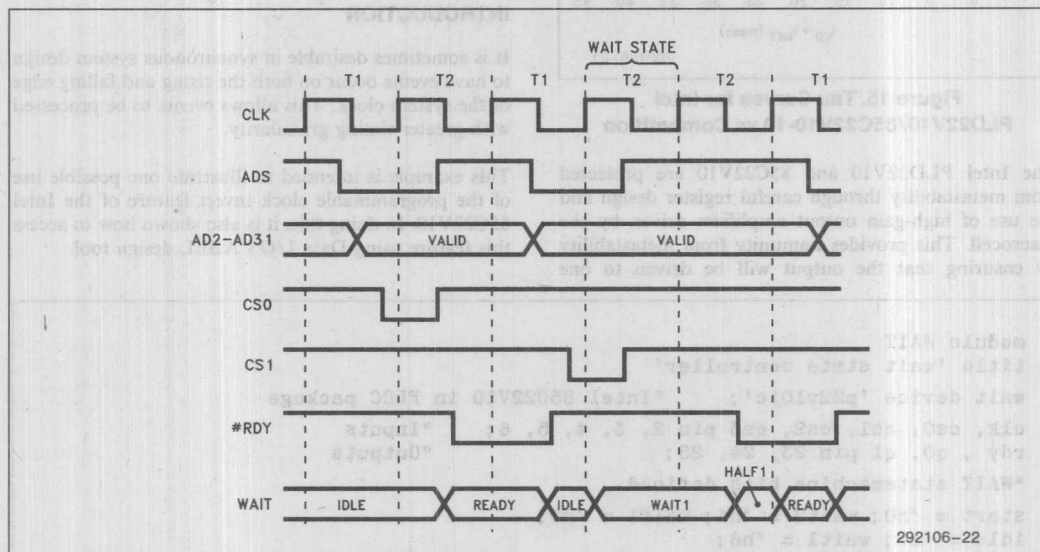


Figure 17. Bus Cycle Diagram

## DESCRIPTION

By watching the Chip Select ( $\overline{\text{CS}}$ ) lines, the WAIT state machine determines the correct number of wait states to insert, the range being from 0–3 in this example.  $\overline{\text{CS}}$  signals are typically decoded from the address signals out of the Intel 486.

Referring to Figure 17, two bus cycles are shown, illustrating only the signals critical to this example. Address signals are not guaranteed to be valid out of a 33 MHz Intel 486 until 16 ns after the rising edge of T1 which is already past the falling edge of T1 (assuming 50% duty cycle, 30 ns clock period). The next opportunity for a clock edge is the rising edge of T2. Using a fast PLD, such as the 7 ns Intel 85C224, the decoding of the address lines into a  $\overline{\text{CS}}$  signal can be done in time to meet the required 7 ns setup time ( $t_{\text{SU}}$ ) into an 85C22V10 that contains the WAIT state machine. Since the first bus cycle shown involves  $\overline{\text{CS0}}$ , it is a zero wait-state

cycle.  $\overline{\text{RDY}}$  is generated by the WAIT state machine on the falling edge of T2, in time to make the 8 ns setup time into the Intel 486 before the rising edge of the next T1. Since  $\overline{\text{RDY}}$  is asserted until the falling edge of T1, ample hold time is provided.

The second bus cycle requires the insertion of a wait-state. This is dictated by the assertion of  $\overline{\text{CS1}}$ . Now, instead of returning a  $\overline{\text{RDY}}$  at the end of one T2 cycle, the WAIT state machine will wait through one additional T2 cycle. Note that for this state sequence WAIT goes through an intermediate "half state" between WAIT1 and READY. This is because the transition from WAIT1 to READY begins on the rising edge of the last T2 and is completed on the falling edge. This is illustrated as a state diagram in Figure 18. When designing a mixed clock phase state machine, care must be taken to account for intermediate half states. Valid state assignments must not conflict with half states or else proper results will not be achieved.

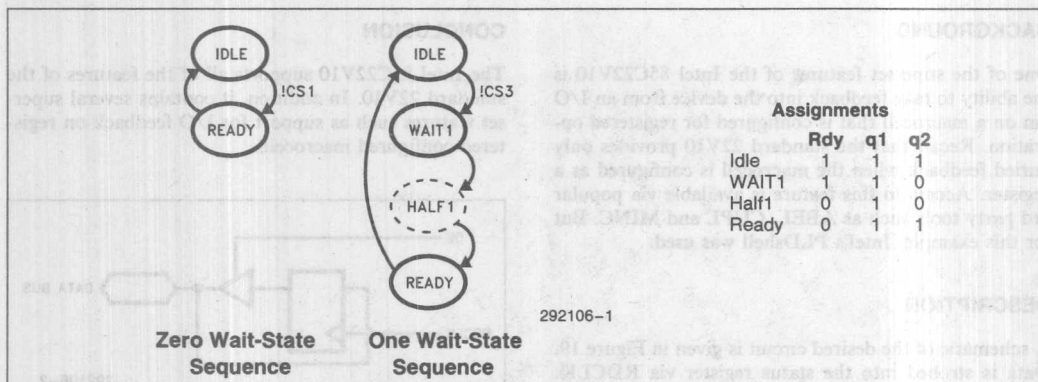


Figure 18. WAIT State Machine State

### CONCLUSION

Obviously this design only utilizes a small portion of the 85C22V10. Ample room exists to add a bus arbiter or additional random logic.

As long as close attention is paid to state assignments, mixed clock phase state machines can be a powerful design tool for meeting tough system timing requirements.

### Design Example #2—Status Register

#### INTRODUCTION

The need often arises in digital system design to store a status data bit or bits in a register for later reference. Typically a bi-directional databus is used for reading and writing this data.

3

## BACKGROUND

One of the superset features of the Intel 85C22V10 is the ability to take feedback into the device from an I/O pin on a macrocell that is configured for registered operation. Recall that the standard 22V10 provides only buried feedback when the macrocell is configured as a register. Access to this feature is available via popular 3rd party tools such as ABEL, CUPL and MINC. But for this example, Intel's PLDshell was used.

## DESCRIPTION

A schematic of the desired circuit is given in Figure 19. Data is strobed into the status register via RDCLK. When it is desired to read back the value of the data, this process is controlled by OE. Up to 10 such data registers may be implemented on a single 85C22V10 with separate OE control for each.

## CONCLUSION

The Intel 85C22V10 supports all of the features of the standard 22V10. In addition, it contains several superset features such as support for I/O feedback on registered configured macrocells.

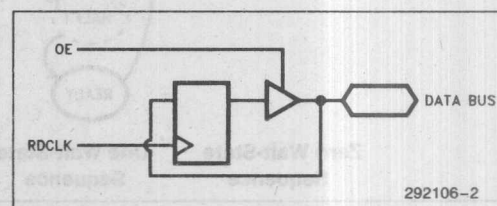


Figure 19. Status Register

```
chip STATUS_REG 85C22V10

; Inputs
PIN RDCLK
PIN CS1
PIN OE

; BIDIRECTIONAL PIN

PIN STAT REGISTERED PINFBK      ; Registered output, pin feedback

EQUATIONS

STAT = STAT                      ; Register input tied to output
STAT.CLKF = RDCLK
STAT.TRST = OE
```

Figure 20. PLDshell PLUS Design File



June 1988

# Implementing a PS/2 POS Using the 5AC312 EPLD

**PEDRO VARGAS**  
PROGRAMMABLE LOGIC APPLICATIONS

Order Number: 292047-001

# IMPLEMENTING A PS/2 POS USING THE 5AC312 EPLD

on the adapter location, the ID of the  
first three POS registers (1, 2, 3) is  
used to identify the adapter ID and the  
required because they provide the adapter ID and the  
adapter enable/disable function necessary during setup  
and error checking. In fact, the way that the system  
uses POS is as follows:

1. The system selects the adapter to be placed in setup  
mode by driving its CD SETUP signal active.
2. The adapter is identified by reading two ID bytes  
from POS 0 and POS 1 (HEX 00 and 01).
3. The adapter is disabled by writing 0 to POS 2  
(HEX 00).
4. If implemented, Option Select Data is written to  
POS 3, 4, 5.
5. The adapter is enabled by writing "1" to POS 2.
6. The adapter is out of setup mode when the system  
drives the CD SETUP signal inactive.

The system hardware implementation of POS is shown  
used in IBM technical documents, but the details are  
left up to each designer.

## ADAPTER REQUIREMENTS

The adapter used for this design is an Intel single-chip  
non card that incorporates two registers connected by a  
bus. Since a processor only one function, there was  
no need to implement the POS Option Select Data.  
(These POS bytes are used with other function adapt-  
ers that do more than one task and reside in the system  
with similar adapters.) In this case, the only require-  
ment was to provide the ID bytes and the enable/dis-  
able features, which are done with POS registers 0, 1,  
and 2. Figure 1 shows the POS register layout and the  
typical POS hardware implementation as suggested by  
IBM. Table 1 defines the POS registers.

## CONTENTS

	PAGE
INTRODUCTION	3-90
PS/2 MICRO CHANNEL*	3-90
POS REQUIREMENTS	3-90
ADAPTER REQUIREMENTS	3-90
5AC312 EPLD	3-92
DEVICE DESCRIPTION	3-92
5AC312 POS IMPLEMENTATION	3-96
RESOURCE ALLOCATION	3-96
DESIGN FILES	3-98
SUMMARY	3-98
ACKNOWLEDGEMENTS	3-98

## PS/2 MICRO CHANNEL

One of the main features in the PS/2 models is the capa-  
bility to do both in and adapter configuration with soft-  
ware instead of hardware. The feature, called POS  
(Programmable Option Select), eliminates the need for  
switches on the motherboard and adapters to replace  
them with programmable registers. The idea is rather  
simple: remove boards and manually setting switches,  
then configure boards and manually setting switches.  
All configuration information is located in files and can  
be read or written to the motherboard or to the adap-  
ter through the Micro Channel. The motherboard and  
each connector on the Micro Channel has a unique sig-  
nal called CD SETUP that initiates a setup mode when  
it is active. Only one connector at a time can be in the  
setup mode, which provides an organized way to per-  
form installation.

## INTRODUCTION

The introduction of the IBM® PS/2 (Personal System/2\*) models and the innovative Micro Channel® has provided numerous opportunities to develop creative interface solutions. Although the interface requirements are new, the designer is faced with making a familiar choice: Use discrete chips (SSI/MSI), incorporate a PLD, or go for the custom IC solution.

In the past, using TTL on the PC/XT/AT bus was often a good choice, but the reduced size of the PS/2 adapters ("plug in boards") increases the cost of board space dramatically. The custom chip solution is probably the best for companies that have a well-defined product, large volumes, and can afford the cost of the chip development. The third choice, using a PLD, is one that has not been popular in PC bus interfacing due to the limited function and performance of most PLDs.

The Intel 5AC312 is a third-generation EPLD that gives designers the resources needed to interface to buses like the Micro Channel. In addition, it provides two benefits not completely provided by either of the other two choices; high integration, and re-programmability. The rest of this application note contains a detailed presentation of a basic POS (Programmable Option Select) implementation for the PS/2 Micro Channel that is done with the 5AC312 EPLD.

## PS/2 MICRO CHANNEL

One of the best features in the PS/2 models is the capability to do system and adapter configuration with software instead of hardware. This feature, called POS (Programmable Option Select), eliminates the need for switches on the motherboard and adapters by replacing them with programmable registers. The idea is, rather than removing boards and manually setting switches, all configuration information is located in files and can be read or written to the motherboard or to the adapters through the Micro Channel. The motherboard and each connector on the Micro Channel has a unique signal called -CD SETUP that initiates a setup mode when it is active. Only one connector at a time can be in the setup mode, which provides an organized way to perform initialization.

## POS REQUIREMENTS

Each adapter must implement POS with eight registers. Depending on the adapter function, not all of them need to be used. The first three (POS registers 0,1,2) are required because they provide the adapter ID and the adapter enable/disable function necessary during setup and error checking. In brief, the way that the system uses POS is as follows:

1. The system selects the adapter to be placed in setup mode by driving its -CD SETUP signal active.
2. The adapter is identified by reading two ID bytes from POS 0 and POS 1 (HEX 100 and 101).
3. The adapter is disabled by writing "0" to POS 2 (HEX 102).
4. If implemented, Option Select Data is written to POS 3, 4, 5.
5. The adapter is enabled by writing "1" to POS 2.
6. The adapter is out of setup mode when the system drives the -CD SETUP signal inactive.

The actual hardware implementation of POS is summarized in IBM technical documents, but the details are left up to each designer.

## ADAPTER REQUIREMENTS

The adapter used for this design is an Intel single-function card that incorporates two modems controlled by a 80C186. Since it performs only one function, there was no need to implement the POS Option Select bytes. (These POS bytes are used with multi-function adapters that do more than one task and reside in the system with similar adapters.) In this case, the only requirements were to provide the ID bytes and the enable/disable features, which are done with POS registers 0,1, and 2. Figure 1 shows the POS register layout and the typical POS hardware implementation as suggested by IBM. Table 1 defines the POS registers.

\*IBM, Personal System/2 and Micro Channel are trademarks of International Business Machines Corporation.

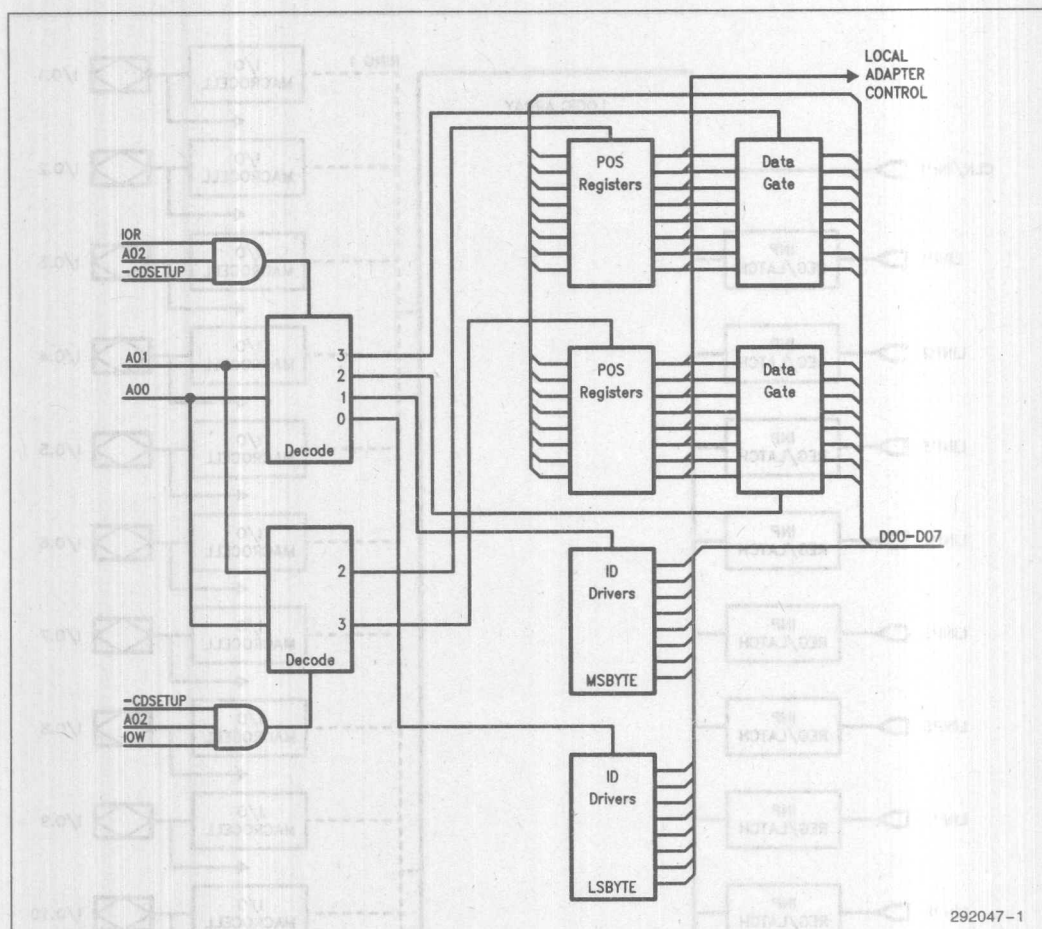


Figure 1. Typical Adaptor Implementation of POS

Table 1. POS I/O Address Decode

Address (hex)	Register	-CD SETUP	Address Bit			Function
			A2	A1	A0	
0100	POS Register 0	0	0	0	0	Adapter Identification Byte (Least Significant Byte)
0101	POS Register 1	0	0	0	1	Adapter Identification Byte (Most Significant Byte)
0102	POS Register 2	0	0	1	0	Option Select Data (Byte 1)*
0103	POS Register 3	0	0	1	1	Option Select Data (Byte 2)
0104	POS Register 4	0	1	0	0	Option Select Data (Byte 3)
0105	POS Register 5	0	1	0	1	Option Select Data (Byte 4)*
0106	POS Register 6	0	1	1	0	Subaddress Extension (Least Significant Byte)
0107	POS Register 7	0	1	1	1	Subaddress Extension (Most Significant Byte)

\*These bytes contain one or more bits with specific value assignments.



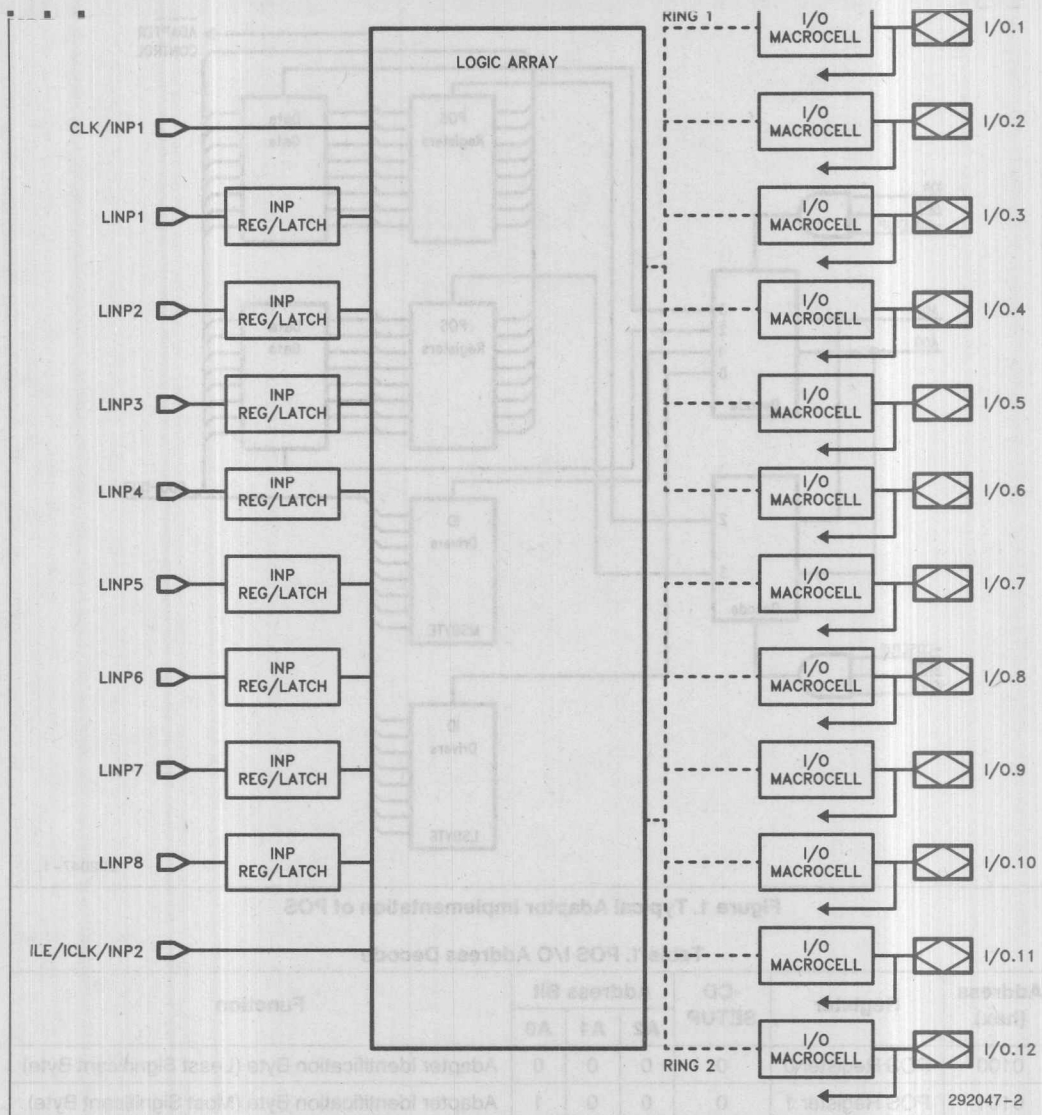


Figure 2. 5AC312 Architecture

## 5AC312 EPLD

With 12 macrocells and a host of other features, the 5AC312 is Intel's newest EPLD. The device is based on the same CHMOS process used in other Intel devices. This EPLD provides an abundant feature set, but its strength lies in being able to efficiently implement one very important function missing from most PLDs: register-logic-register functions.

## DEVICE DESCRIPTION

The 5AC312 (Figure 2) contains 12 macrocells with programmable outputs and inputs. A macrocell is the basic block associated with each output register within the EPLD. The 5AC312 has the following features:

- 12 I/O macrocells with dual feedback for implementing buried registers.

- 8 programmable inputs that can be configured as latches, registers, or flow through inputs. These can be clocked synchronously or asynchronously.
- Product term allocation on each macrocell.
- 2 product terms on all macrocell control signals.
- 2 multi-function pins; a CLK/INPUT and a ILE/ICLK/INPUT.
- 40 MHz operation.

The 5AC312 provides three major benefits that are especially important to designers working on bus interfaces:

1. The availability of input latches (Figure 3) makes it easy to synchronize bus control signals synchronously or asynchronously. The latches can be clocked as a group of 8 or individually, as is quite common on most buses. Input latches also make state machine designs more reliable. Since buses are prone to glitches and other transients, the ability to hold the inputs stable while transitioning through states makes the difference between a clean and a jittery state machine.
2. Product Term Allocation (Patent Pending) brings a new concept to the Intel EPLD family and makes the 5AC312 unique among PLDs. This feature means that the designer can implement large designs that contain as few as zero or as many as 16 product

terms per macrocell (Figure 4). Product Term Allocation takes place in two rings of six macrocells. Within each ring (Figure 2), individual macrocells can allocate p-terms to/from adjacent macrocells. This is a real benefit in bus decoding where intermediate signals can have few or many p-terms all within the same logic function. Most designers that use PLDs have at least one horror story of a design that required 10 or more p-terms and a device that could only provide 8.

3. A flexible output structure is a must for efficient bus interfacing, which quite commonly requires lots of I/O and complex control signals. The 5AC312 meets these demands head-on with dual-feedback paths and two p-terms per control signal on all I/O macrocells. This means that certain functions, like state machines, can be buried and a pin won't be wasted because it can be used as an additional input. Also, output enables and register operations are frequently generated by a combination of memory, I/O, read, and write strobes. Many times these control signals require two p-terms or the equivalent of an external read/write multiplexer. Prior to the 5AC312, the only way to implement this in PLDs was to waste a macrocell to inefficiently provide this function. Figure 5 shows the macrocell structure and details this third benefit.

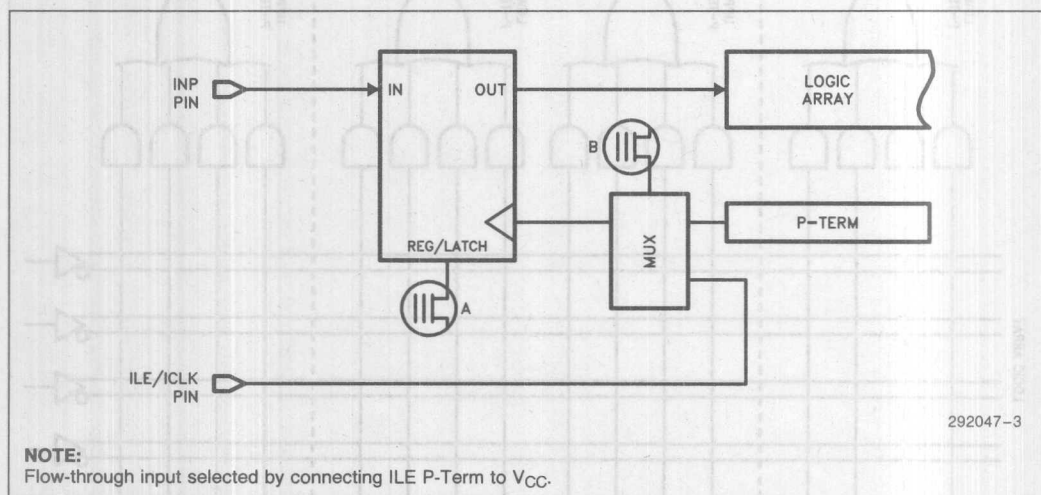
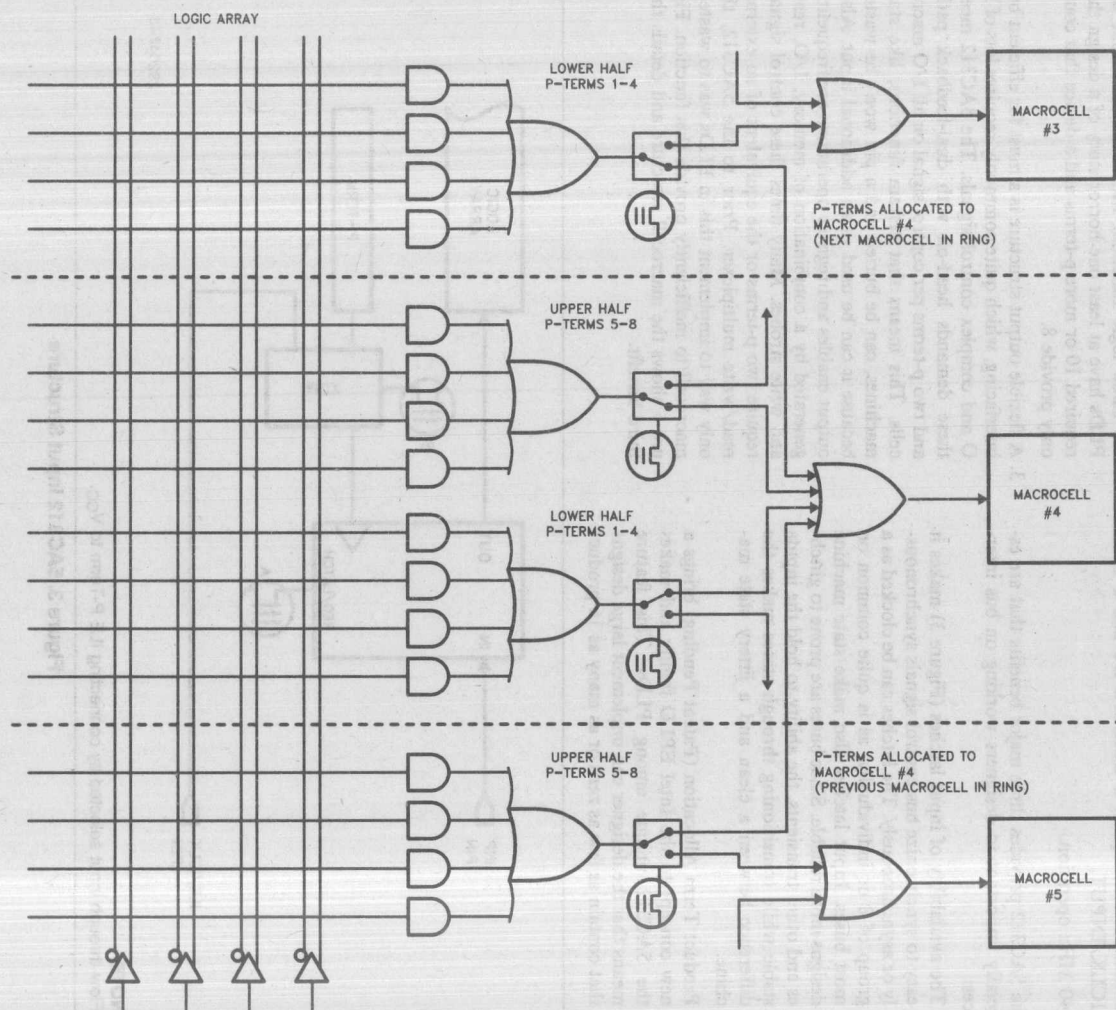
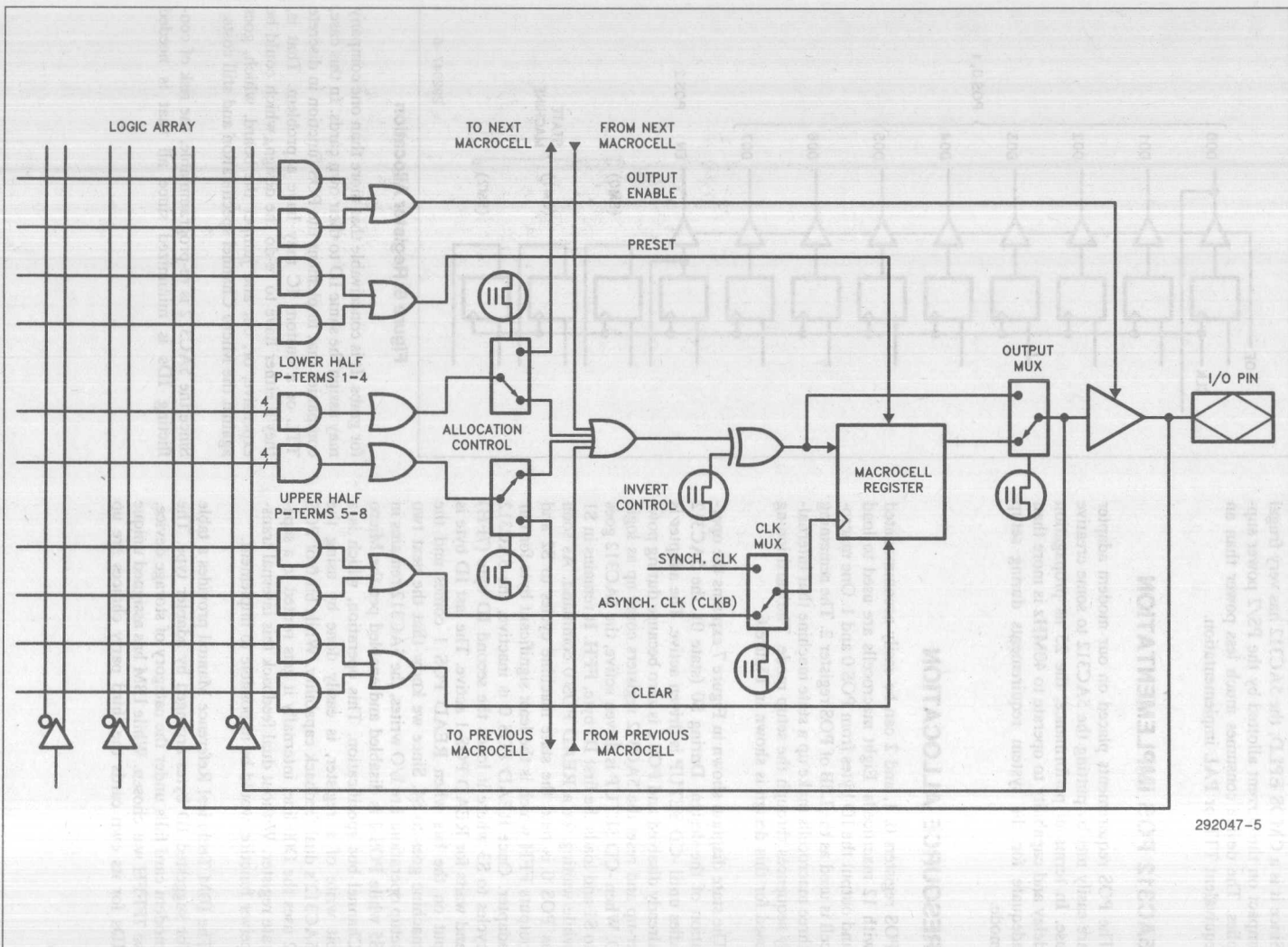


Figure 3. 5AC312 Input Structure



292047-4

Figure 4. Product Term Allocation (8 + 4 + 4)





plies. The device consumes much less power than an equivalent TTL or PAL implementation.

## 5AC312 POS IMPLEMENTATION

The POS requirements placed on our modem adapter are easily met by putting the 5AC312 to some creative use. In terms of performance, the 25 ns propagation delay and capability to operate to 40MHz is more than adequate for the system requirements during setup mode.

## RESOURCE ALLOCATION

POS registers 0, 1, and 2 can be easily accommodated with 12 macrocells. Eight macrocells are used to load and output the ID bytes from POS 0 and 1. One macrocell is used as the LSB of POS register 2. The remaining three macrocells make up a state machine that internally sequences through the setup mode. The partitioning used for this design is shown in Figure 6.

The state diagram shown in Figure 7 explains the operation of the design. During S0 (state 0), the 5AC312 idles until -CD SETUP is driven active. The adapter is already disabled and POS 2 is zero because during power-up and reset the 5AC312 registers come up as logic 0. When -CD SETUP is driven active, the 5AC312 goes to S1 and loads the first ID byte, FFH. It remains in S1 while waiting for a READ POS 0 command. As soon as POS 0 is read the state machine cycles to S2 and outputs FFH which is the least significant byte for our adapter. Once READ POS 0 is inactive, the 5AC312 cycles to S3 where it loads the second ID byte (7FH) and waits for READ POS 1 active. The last ID byte is put on the bus when READ POS 1 comes and the machine goes to S4. Since we know that the next two setup operations are I/O writes, the 5AC312 remains in S5 while POS 2 is disabled and enabled per the Micro Channel bus specification. This operation, which is a bit write of a register, is easily done by using the 5AC312's dual feedback capability. While bit 0 of POS 2 uses the D00 line, internally it gets routed to a separate register. Without dual feedback this internal transceiver function would be impossible to implement.

The IBM Technical Reference Manual provides a table for suggested ID bytes arranged by adapter type. The modem card falls under the category of storage device, so 7FFFH was chosen. While IBM has assigned unique IDs for its own cards the third party choices are up

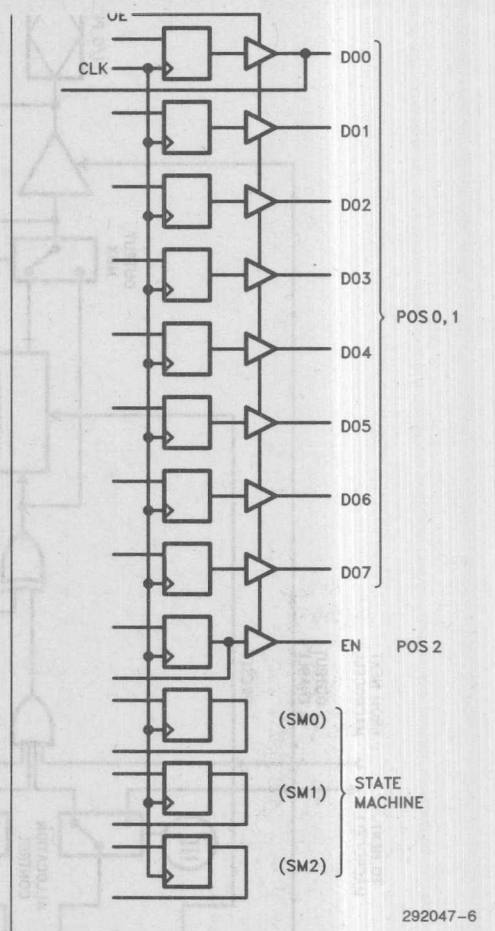
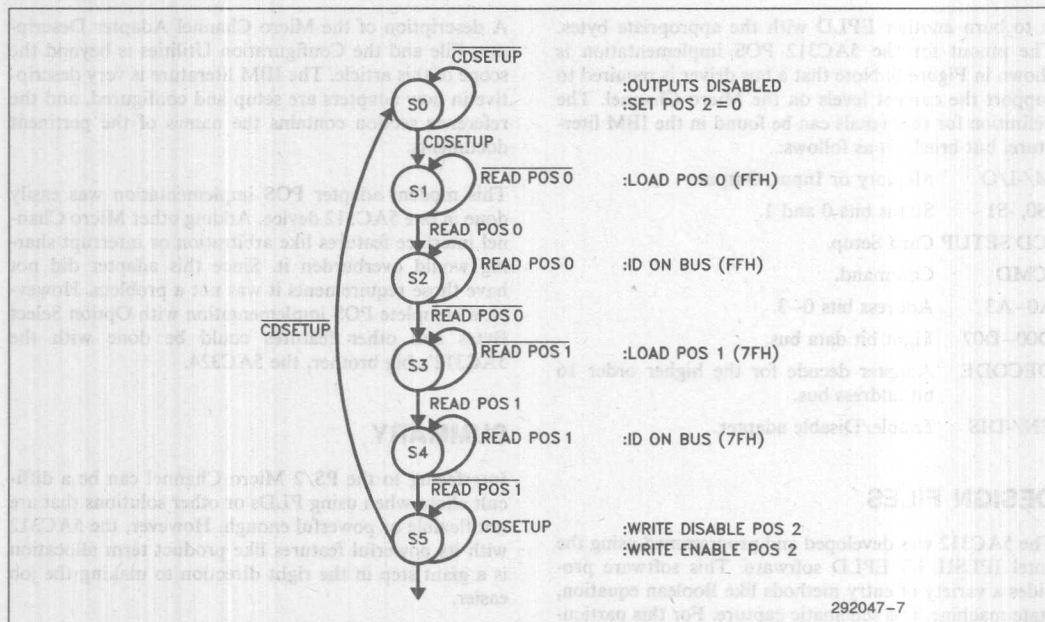


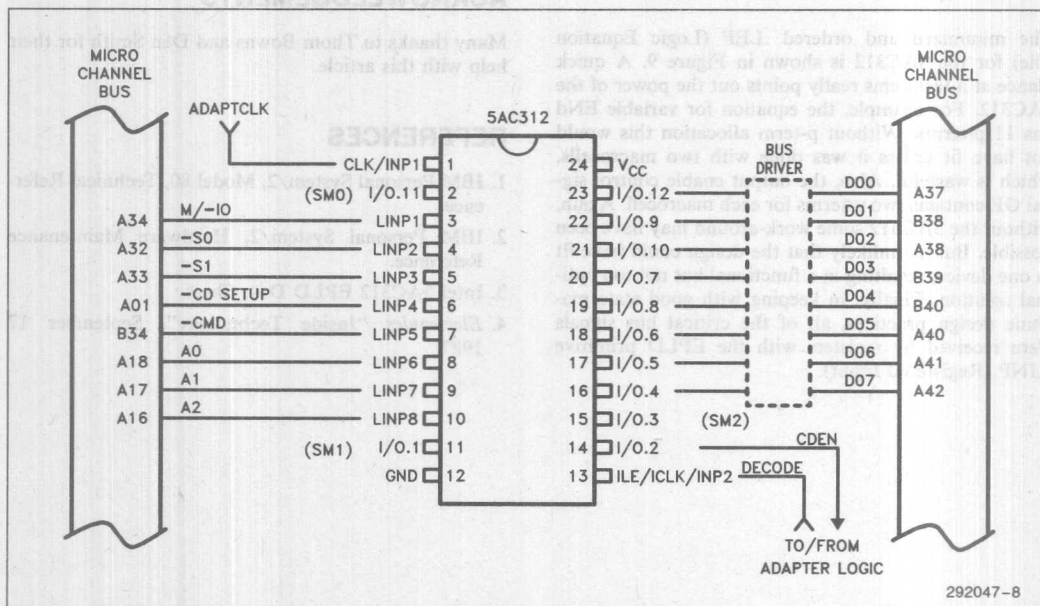
Figure 6. Register Allocation

for grabs. It is conceivable that more than one company may assign the same ID to their own cards. In this case, companies that implement the POS function in discrete TTL or a custom IC may have a problem. That is, they'll either have to re-do the design, which could be expensive, or, cut and jumper the board, which, goes against the Micro Channel specification and still costs.

Since the 5AC312 is re-programmable, the risk of conflicting IDs is minimized since all that is needed



3



is to burn another EPLD with the appropriate bytes. The pinout for the 5AC312 POS implementation is shown in Figure 8. Note that a bus driver is required to support the current levels on the Micro Channel. The definition for the signals can be found in the IBM literature, but briefly is as follows:

M/-I/O	Memory or Input/Output.
-S0, -S1	Status bits 0 and 1.
-CD SETUP	Card Setup.
-CMD	Command.
A0-A3	Address bits 0-3.
D00-D07	Eight bit data bus.
DECODE	Adapter decode for the higher order 16 bit address bus.
EN/-DIS	Enable/Disable adapter.

## DESIGN FILES

The 5AC312 was developed and programmed using the Intel IPLSII 1.5 EPLD software. This software provides a variety of entry methods like Boolean equation, state machine, and schematic capture. For this particular design, equation entry was the most convenient since the implementation was done in one IC.

The minimized and ordered .LEF (Logic Equation File) for the 5AC312 is shown in Figure 9. A quick glance at a few items really points out the power of the 5AC312. For example, the equation for variable END has 11 p-terms. Without p-term allocation this would not have fit unless it was done with two macrocells, which is wasteful. Also, the output enable control signal OE contains two p-terms for each macrocell. Again, without the 5AC312 some work-around may have been possible. But its unlikely that the design could have fit in one device, resulting in a functional but not too optimal solution. Finally, in keeping with good state machine design practices, all of the critical bus signals were received by registers with the EPLD primitive RINP (Registered Input).

A description of the Micro Channel Adapter Description File and the Configuration Utilities is beyond the scope of this article. The IBM literature is very descriptive in how adapters are setup and configured, and the reference section contains the names of the pertinent documents.

This modem adapter POS implementation was easily done in one 5AC312 device. Adding other Micro Channel interface features like arbitration or interrupt sharing would overburden it. Since this adapter did not have those requirements it was not a problem. However, a complete POS implementation with Option Select Bytes and other features could be done with the 5AC312's big brother, the 5AC324.

## SUMMARY

Interfacing to the PS/2 Micro Channel can be a difficult chore when using PLDs or other solutions that are not flexible or powerful enough. However, the 5AC312 with its powerful features like product term allocation is a giant step in the right direction to making the job easier.

## ACKNOWLEDGEMENTS

Many thanks to Thom Bowns and Dan Smith for their help with this article.

## REFERENCES

1. IBM Personal System/2, Model 80, Technical Reference.
2. IBM Personal System/2, Hardware Maintenance Reference.
3. Intel 5AC312 EPLD Data Sheet.
4. *Electronics* "Inside Technology", September 17 1987.

```

THOM BOWNS
INTEL
DECEMBER 4, 1987
1
001
5AC312
Implements POS for the PS/2 using a 5AC312.
LEF Version 1.5 Baseline 4.1i 21 Nov 1987
OPTIONS: TURBO=ON
PART:
    5AC312
INPUTS:
    MIO@3, nS0@4, nS1@5, nCDSETUP@6, nCMD@7, A0@8, A1@9, A2@10, DECODE@13,
    CLK@1, D0@23
OUTPUTS:
    D00@23, D01@22, D02@21, D03@20, D04@19, D05@18, D06@17, D07@16, EN@14,
    SM0@2, SM1@11, SM2@15
NETWORK:
    IRE = CLKB(CLK)
    CLK = INP(CLK)
    MIO = RINF(MIO, IRE, GND, GND)
    nS0 = RINF(nS0, IRE, GND, GND)
    nS1 = RINF(nS1, IRE, GND, GND)
    nCDSETUP = RINF(nCDSETUP, IRE, GND, GND)
    nCMD = RINF(nCMD, IRE, GND, GND)
    A0 = RINF(A0, IRE, GND, GND)
    A1 = RINF(A1, IRE, GND, GND)
    A2 = RINF(A2, IRE, GND, GND)
    DECODE = INP(DECODE)
    D0 = INP(D0)
    D00 = RINF(D00d, CLK, GND, GND, OE)
    D01 = RINF(D01d, CLK, GND, GND, OE)
    D02 = RINF(D02d, CLK, GND, GND, OE)
    D03 = RINF(D03d, CLK, GND, GND, OE)
    D04 = RINF(D04d, CLK, GND, GND, OE)
    D05 = RINF(D05d, CLK, GND, GND, OE)
    D06 = RINF(D06d, CLK, GND, GND, OE)
    D07 = RINF(D07d, CLK, GND, GND, OE)
    EN, EN = RINF(ENd, CLK, GND, GND, VCC)
    SM0 = NORF(SM0d, CLK, GND, GND)
    SM1 = NORF(SM1d, CLK, GND, GND)
    SM2 = NORF(SM2d, CLK, GND, GND)

```

292047-9

Figure 9. POS Design File



EQUATIONS:

```

SM2d = SM2' * SM1 * SM0 * nCDSETUP' * MIO' * ns0' * A2' * A1' * DECODE *
      ns1
      + SM2 * SM1' * nCDSETUP';

SM1d = SM2' * SM1' * SM0 * nCDSETUP' * MIO' * ns0' * A2' * A1' * DECODE *
      ns1
      + SM2' * SM1 * SM0' * nCDSETUP'
      + SM2' * SM1 * nCDSETUP' * DECODE'
      + SM2' * SM1 * nCDSETUP' * A1
      + SM2' * SM1 * nCDSETUP' * A2
      + SM2' * SM1 * nCDSETUP' * ns1'
      + SM2' * SM1 * nCDSETUP' * ns0
      + SM2' * SM1 * nCDSETUP' * MIO;

SM0d = (SM2' * SM0 * MIO' * ns0' * ns1 * A2' * A1' * DECODE
      + SM2 * SM0' * MIO' * ns0' * ns1 * A2' * A1' * DECODE
      + SM1 * MIO' * ns0' * ns1 * A2' * A1' * DECODE
      + SM2 * SM1
      + nCDSETUP)';

End = D0 * MIO' * A2' * A1 * A0' * DECODE * SM2 * SM1' * SM0 * ns1' * ns0
      + ns0' * EN
      + ns1 * EN
      + SM0' * EN
      + SM1 * EN
      + SM2' * EN
      + DECODE' * EN
      + A0 * EN
      + A1' * EN
      + A2 * EN
      + MIO * EN;

D07d = SM2' * SM0'
      + SM2' * SM1';

D06d = (SM2 * SM1)';

D05d = (SM2 * SM1)';

D04d = (SM2 * SM1)';

D03d = (SM2 * SM1)';

D02d = (SM2 * SM1)';

D01d = (SM2 * SM1)';

OE = SM2 * SM1' * SM0' * MIO' * ns0' * A2' * A1' * DECODE * ns1
      + SM2' * SM1 * SM0' * MIO' * ns0' * A2' * A1' * DECODE * ns1;

D00d = (SM2 * SM1)';

```

END\$

292047-10

Figure 9. POS Design File (Continued)

December 1988

3

# Designing with the 5AC312/5AC324 EPLDs

**DAVID BICKEL**  
PROGRAMMABLE LOGIC APPLICATIONS  
INTEL CORPORATION

Order Number: 292049-001

# DESIGNING WITH THE 5AC312/5AC324 EPLDS

## CONTENTS

PAGE

INTRODUCTION .....	3-103
PROGRAMMABLE INPUTS .....	3-103
MACROCELL STRUCTURE .....	3-103
MULTIPLE P-TERMS .....	3-103
P-TERM ALLOCATION .....	3-103
DUAL FEEDBACK .....	3-108
POWER-ON CHARACTERISTICS .....	3-108
POWER-DOWN MODE .....	3-108
EXAMPLE .....	3-109
SUMMARY .....	3-109

## INTRODUCTION

The Intel 5AC312 EPLD (Erasable Programmable Logic Device) was developed to break down certain existing PLD architectural barriers and meet increased performance needs. The Intel 5AC312 EPLD was designed by EPLD users with direct input from system designers. In the design process, emphasis was placed first on gate utilization, and then on density.

This application note highlights the advanced architecture and features of the 5AC312 EPLD and shows the benefits of designing with this new device over more traditional PLD architectures. These features include enhanced input structure with register/latch option on all input pins (synchronous or asynchronous operation); user-controllable, software-supported p-term allocation scheme in all macrocells; and multiple p-terms on control functions (asynchronous CLK, SET, RESET, OE).

It should also be noted that the features and information described here also apply to the new 5AC324 EPLD. The 5AC324 is basically a 24 macrocell version of the 5AC312.

## PROGRAMMABLE INPUTS

The 5AC312 was designed with a highly flexible macrocell and I/O structure allowing the device to implement both combinational and sequential logic functions. The enhanced input structure not only allows the device to latch and hold incoming data, but also to implement register-combinational-register logic to easily accommodate state machine designs. Figure 1 shows a global view of the 5AC312 architecture.

The 5AC312 is equipped with 8 user-programmable input structures that can each be configured to work in one of five modes: 1) synchronous D-type register, 2) asynchronous D-type register, 3) synchronous D-type latch, 4) asynchronous D-type latch, and 5) flow-through input. Each input can be configured independently of the others. The desired configuration is implemented through the programming of EPROM architecture control bits by the logic compiler under user-control.

## MACROCELL STRUCTURE

The 5AC312 also has a unique macrocell array structure that allows for user-controllable, software-supported product term allocation in each of its 12 macrocells. Each of the 12 macrocells also has a dual feedback option with independent feedback and I/O paths. Each macrocell has 16 product terms, 8 of which control the OE, PRESET, ASYNCHRONOUS CLOCK, and

CLEAR signals (2 p-terms per signal). The other 8 feed the data input to the macrocell and are split into two groups of four (upper half and lower half). See Figure 2. Each group of four can be allocated to an adjacent macrocell if needed.

As shown in Figure 1, the 12 macrocells of the 5AC312 are further divided into two "rings" with 6 macrocells per ring. Allocation of p-terms to adjacent macrocells can occur with a given ring. See Figure 3 for p-term allocation scheme.

Each macrocell register in the 5AC312 is also equipped with an asynchronous PRESET signal. The PRESET function can be constructed in more traditional architectural devices such as the 5C060 and 5C090 using combinational logic and feedback, however two macrocells are consumed in the process. To illustrate this difference, compare Figure 2 to the implementation shown in Figure 4. The PRESET function would require additional macrocells in traditional architectures if it were expanded beyond a single p-term.

## MULTIPLE P-TERMS

Multiple p-terms on the control functions (asynchronous CLOCK, PRESET, RESET, and OE) increases the efficiency of the device. Multiplexed I/O is accomplished by controlling the output buffer associated with each macrocell using the 2 p-terms set aside for implementing an OE function. Multiple p-terms create a means to avoid using macrocells for control logic. For example, it would take two macrocells in the 5C060 and 5C090 EPLD to drive the OE line by a 2 p-term signal. To illustrate, compare Figure 2, the 5AC312 macrocell structure, to Figure 5, a diagram of how a two p-term OE signal can be implemented in a 5C060 or 5C090 EPLD.

## P-TERM ALLOCATION

P-term allocation allows for more efficient use of p-terms and thus increased device utilization by raising the number of p-terms per macrocell to 16. P-term allocation, where p-terms are dedicated to certain macrocells, should not be confused with p-term sharing, where several macrocells can actually use the same p-terms. The p-term allocation scheme in all macrocells is user-controllable and software supported, and provides the ability to satisfy designs with large p-term requirements. P-term allocation is ideal for p-term intensive applications such as complex counters or comparators.

P-term allocation in the 5AC312 is used when a design requires one of the 12 macrocells to employ more than 8 p-terms. P-term allocation is simply the transfer



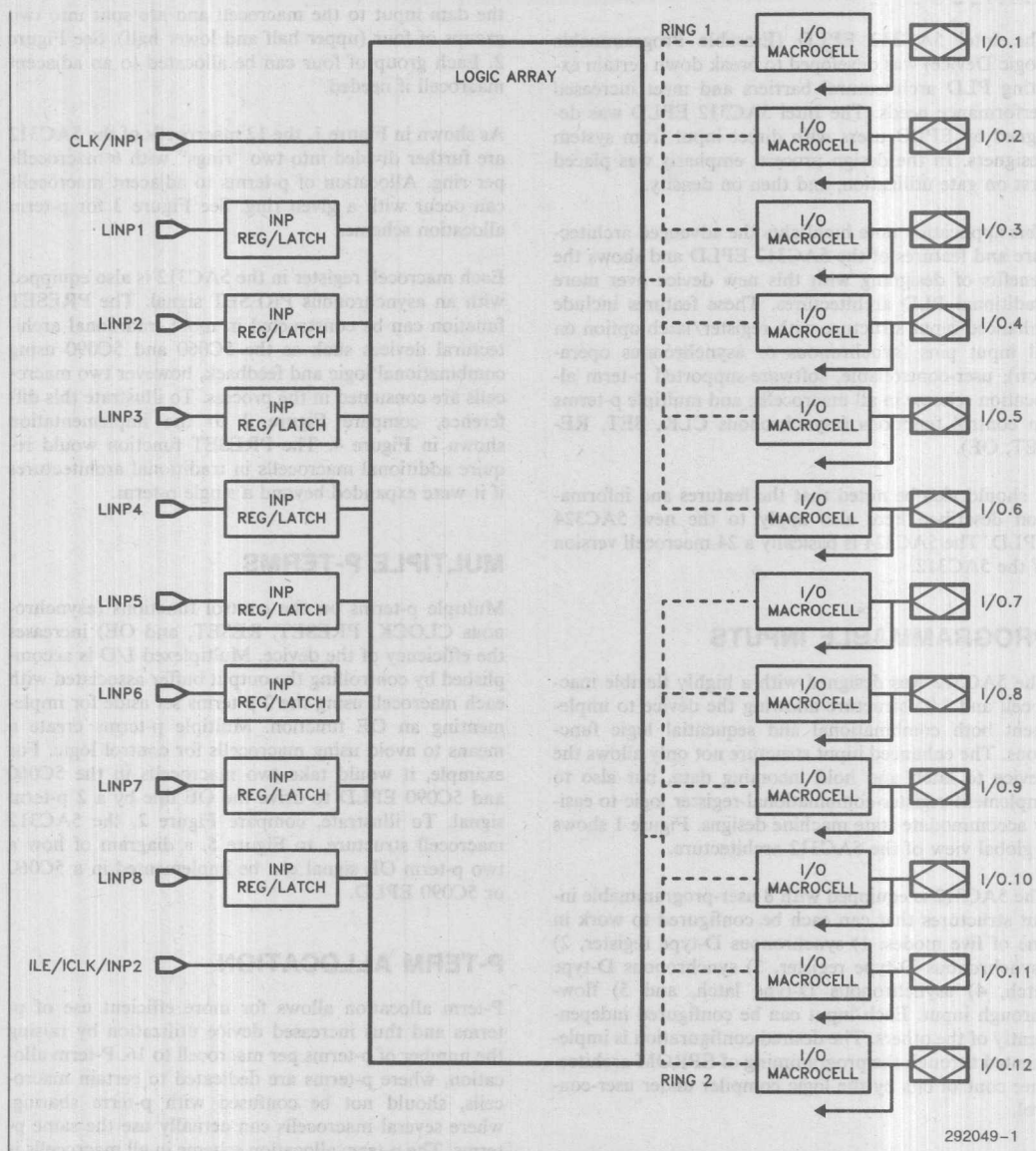


Figure 1. 5AC312 Architecture

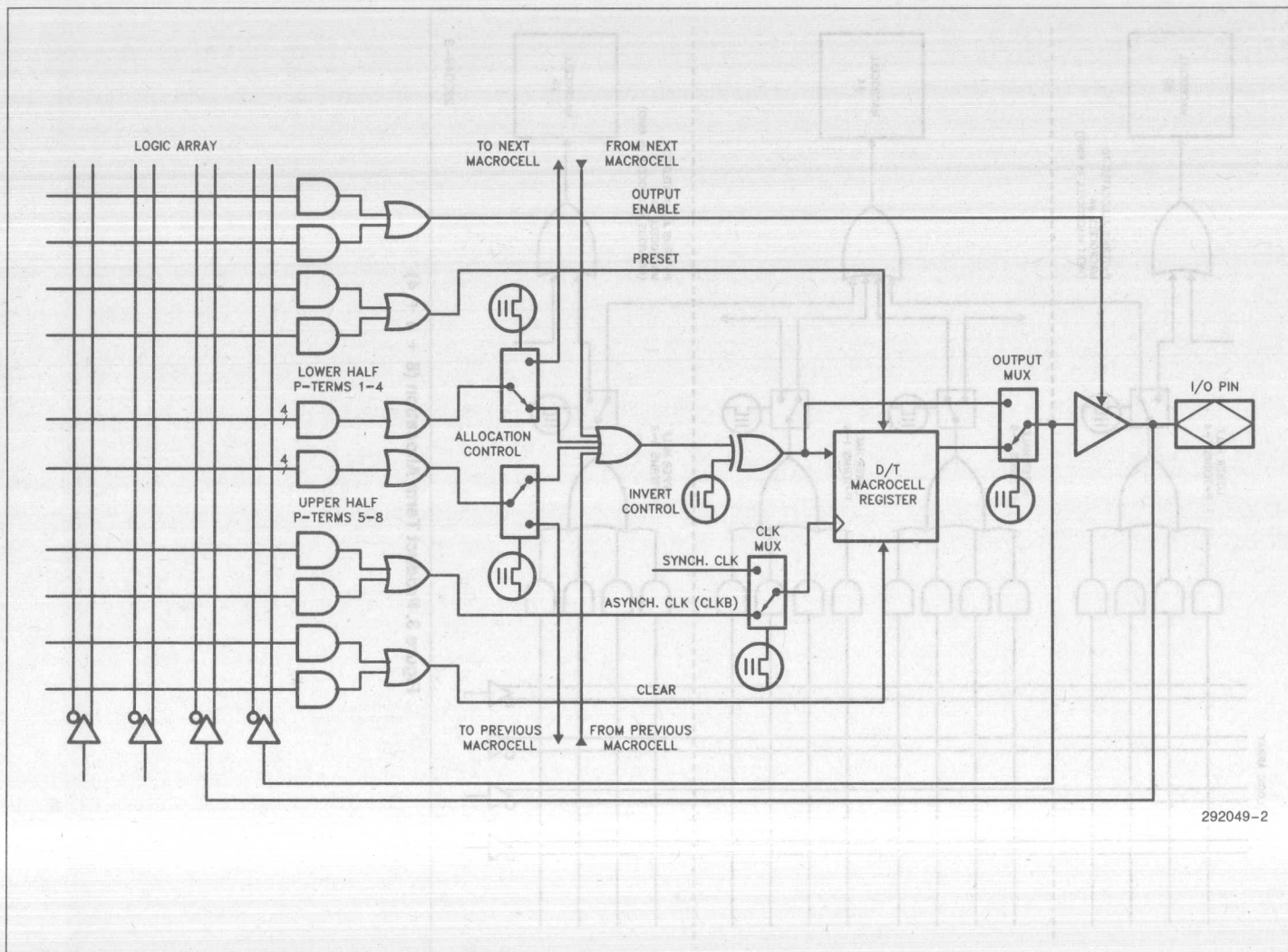


Figure 2. 5AC312 Basic Macrocell Structure

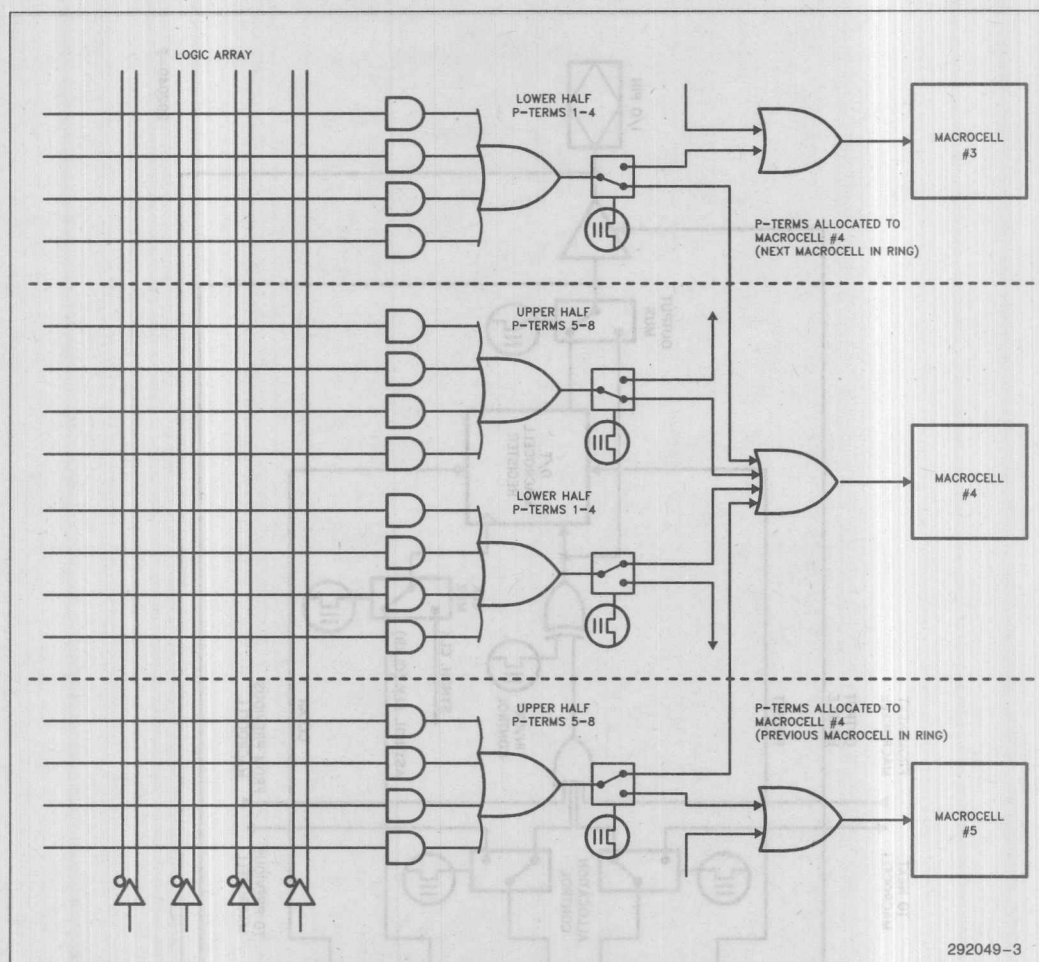
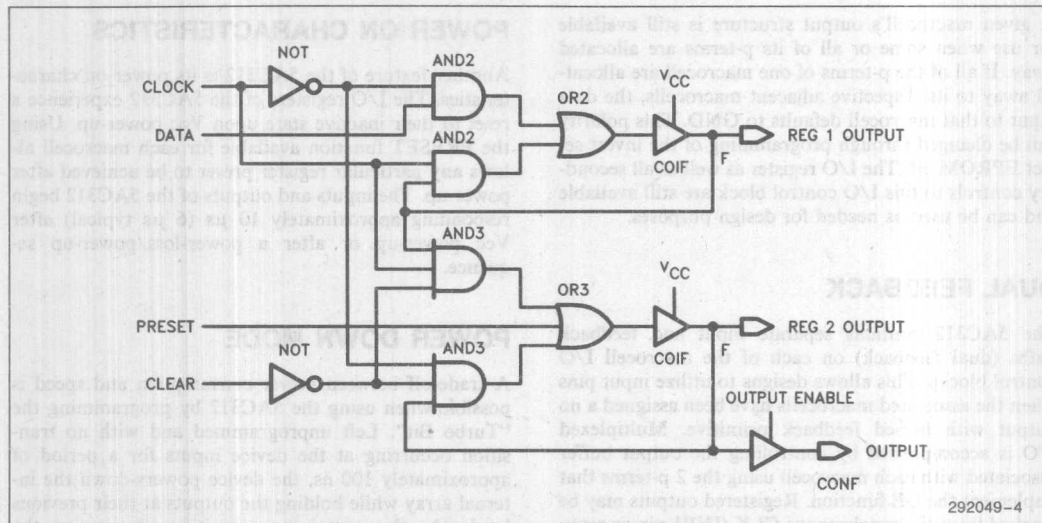


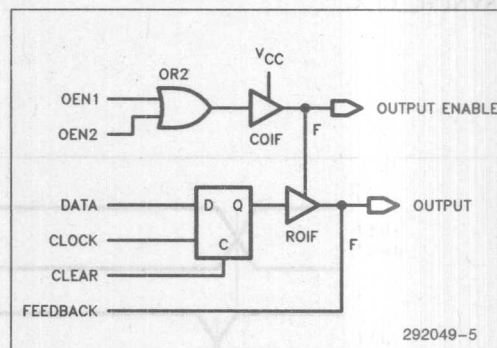
Figure 3. Product Term Allocation (8 + 4 + 4)



**Figure 4. Implementation of D Flip-Flop with Added Preset Function Using Combinational Logic**

of logic resources (p-terms) from areas they are not being utilized to other areas within the chip where they are needed. As shown in Figure 3, each macrocell has the potential to borrow 4 more p-terms to add to the 8 it already has from each of its adjacent macrocells. This increases the maximum number of p-terms per macrocell to 16. Thus, any macrocell within the 5AC312 has the potential to satisfy logic functions requiring between 0 and 16 p-terms.

P-terms can be allocated in a "shift register" mode within each of the two rings of the macrocell; however, allocation of p-terms between rings is not possible. See Table 1 for a listing of adjacent macrocells within p-term allocation rings.



**Figure 5. Implementation of 2 P-Term OE Control Signal**

**Table 1. 5AC312 Product Term Allocation Rings**

Ring 1			Ring 2		
Current Macrocell	Next Macrocell	Previous Macrocell	Current Macrocell	Next Macrocell	Previous Macrocell
1	2	8	7	8	12
2	3	1	8	9	7
3	4	2	9	10	8
4	5	3	10	11	9
5	6	4	11	12	10
6	1	5	12	7	11



for use when some or all of its p-terms are allocated away. If all of the p-terms of one macrocell are allocated away to its respective adjacent macrocells, the data input to that macrocell defaults to GND. This polarity can be changed through programming of the invert select EPROM bit. The I/O register as well as all secondary controls to this I/O control block are still available and can be used as needed for design purposes.

## DUAL FEEDBACK

The 5AC312 contains separate input and feedback paths (dual feedback) on each of the macrocell I/O control blocks. This allows designs to utilize input pins when the associated macrocells have been assigned a no output with buried feedback primitive. Multiplexed I/O is accomplished by controlling the output buffer associated with each macrocell using the 2 p-terms that implement the OE function. Registered outputs may be clocked from the synchronous CLK/INP1 pin or asynchronously clocked by the 2 p-terms available for ASYNCH\_CLK.

Another feature of the 5AC312 is its power-on characteristics. The I/O registers of the 5AC312 experience a reset to their inactive state upon Vcc power-up. Using the PRESET function available for each macrocell allows any particular register preset to be achieved after power-up. The inputs and outputs of the 5AC312 begin responding approximately 10  $\mu$ s (6  $\mu$ s typical) after Vcc power-up or after a power-loss/power-up sequence.

## POWER DOWN MODE

A trade-off between power consumption and speed is possible when using the 5AC312 by programming the "Turbo Bit". Left unprogrammed and with no transition occurring at the device inputs for a period of approximately 100 ns, the device powers-down the internal array while holding the outputs at their previous levels. At the next input transition occurrence, the 5AC312 powers-up the array and reacts to the change in input conditions. If the "Turbo Bit" is programmed, the power-down circuitry is disabled and the device will not power-down even if there are no more transitions. The array power-up sequence requires an additional 20 ns of propagation delay. Power supply current during power-down is no more than 120  $\mu$ A. See Figure 6.

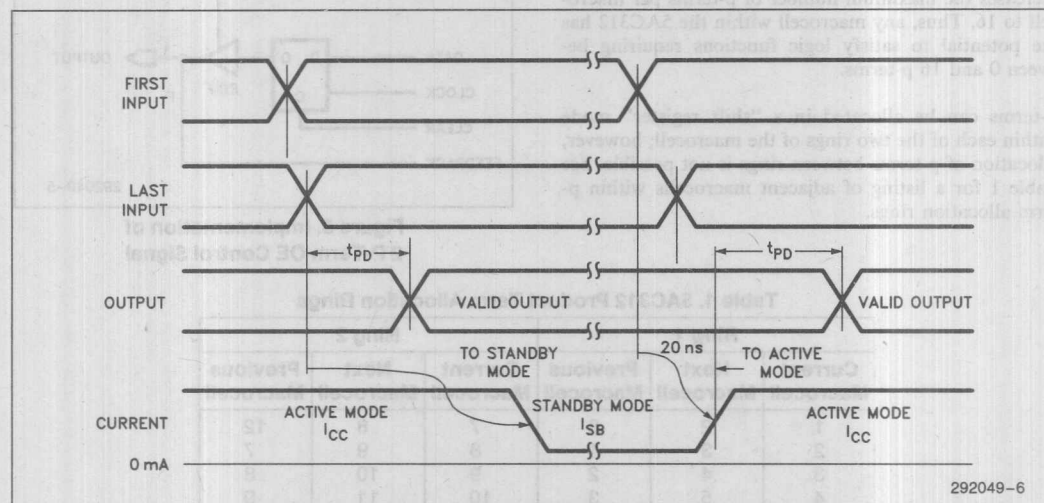


Figure 6. 5AC312 Standby and Active Mode Transitions

## EXAMPLE

An example application for the 5AC312 can be shown by replacing a PAL\* 20R6 and a 374 D type flip-flop in a design due to a power constraint. The same implementation can be achieved consuming less power using one 5AC312 EPLD. Compare Figures 7 and 8. Straight jumpers can be substituted in the PC board where the 374 sits, and since the clock signal is already available on the PAL socket, it can be internally routed to clock the input registers of the 5AC312. The 5AC312 can then be programmed to match the existing pin assignments and therefore require no PC board re-layout. The internal circuitry of the 5AC312 allows the EPLD to act as both a D type flip-flop and a PAL.

## SUMMARY

The 5AC312 EPLD, which uses advanced CMOS EPROM cells as logic control elements instead of polysilicon fuses, represents an innovative device to help overcome the primary limitations of standard PLDs. With its advanced features, proprietary architecture and macrocell structure, the 5AC312 is capable of implementing high performance logic functions more effectively than was previously possible. The p-term allocation scheme is a unique feature, increasing the efficiency of the device immensely. The PRESET signal and 2 p-term control lines are also features giving the 5AC312 added efficiency in many designs.

These same architectural features have been included in the 5AC324 EPLD, making that device ideal for even higher integration applications. Refer to the 5AC324 Data Sheet for details on that device.

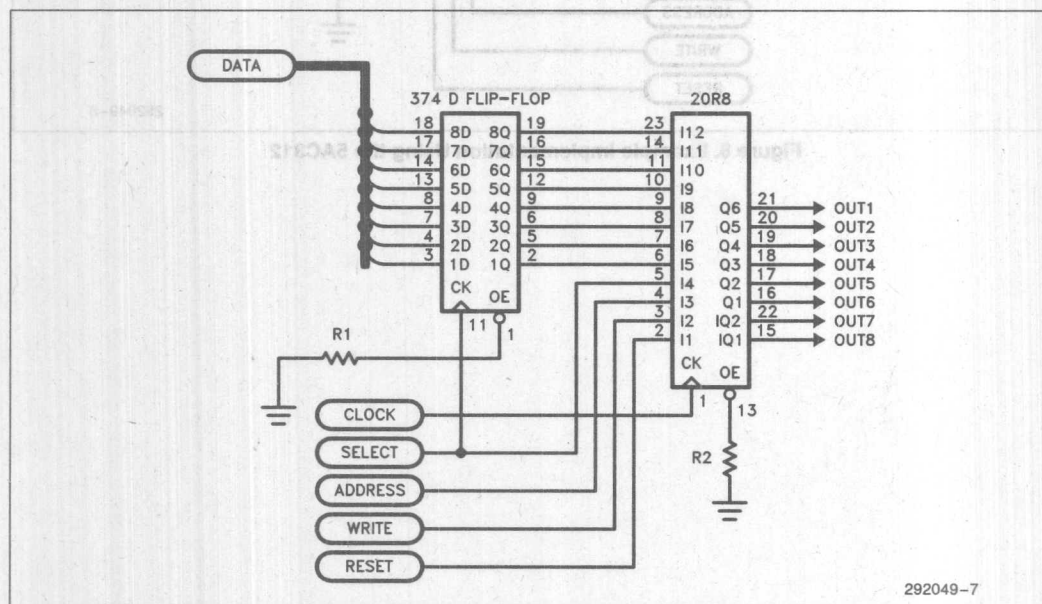


Figure 7. Original Implementation Using a 374 D Flip-Flop and A PAL20R6

\*PAL is a registered trademark of Monolithic Memories, Inc.



## IDEAS FOR DESIGN

A MICRO CHANNEL  
SLAVE-ADAPTER LINK

TODD K. KOELLING

Intel Corp., 1900 Prairie City Rd., Folsom, CA 95630; (916) 351-5788.

Many times neither of the two available implementations for executing a PS/2 Micro Channel slave adapter are optimum solutions. Precustomized PS/2 Micro Channel chips may offer more functionality than necessary for many slave-adapter designs. A standard PAL-and-TTL solution, however, can require too much of the valuable PS/2 adapter-board space.

For the Micro Channel slave-adapter interfaces that are too small for full-custom design, but too large for a PAL implementation, the 5AC324 offers a third solution. In addition to saving board space, the 5AC324 saves power. With a 50-mA  $I_{CC}$ , the CMOS part consumes much less power than its bipolar (or CMOS) PAL substitutes. A 40-pin, 24-macrocell programmable logic device (EPLD), the 5AC324 supplies high integration and includes input latches. Furthermore, designers can customize it for each interface design.

For example, consider an interface for an 8-bit I/O slave (see the figure).

The 5AC324's transparent input latches makes latching incoming address and bus-cycle information convenient and ensures valid data throughout the command cycle. Decoding the latched data generates the I/O and transceiver control outputs, (IORD and IOWR and TOE). The inputs latch on the Address Decode Latch (ADL) signal; the Command (CMD) signal gates the outputs.

For read or write cycles, the 5AC324 EPLD first establishes the transceiver direction with the Transceiver Send/Receive (TS/R) signal. Then, the 5AC324 enables the transceiver (TOE) and generates the read/write strobe (IORD or IOWR) when the command cycle (CMD) becomes active. The latched Micro Channel bus-cycle information signals needed to generate these controls are the memory-I/O cycle (M/I/O), Bus Status ( $\overline{S1}$ ,  $\overline{S0}$ ), Card-Setup (CDSETUP), and Board Select Decode (BDSEL) signals.

Since the CDSFDBK signal must

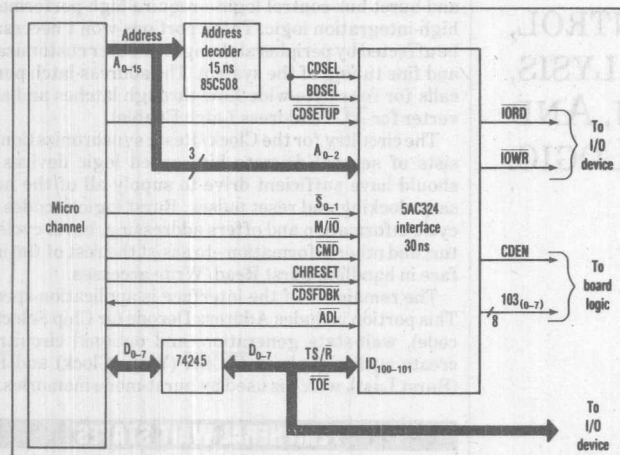
be generated off the unlatched address decode, a second address decode (CDSEL) feeds into the EPLD. The CDSEL signal remains unlatched while BDSEL is latched for the control strobes. The address decoding could be performed inside the 5AC324. However, because that function requires many pins and little logic, an external 8C508 EPLD gives more efficient address decoding.

Besides generating the control signals, the 5AC324 also houses a significant amount of programmable option-select (POS) logic. The 5AC324 implements the POS IDs (POS 100 and 101), Card-Enable register (POS 102, bit 0), and option-select databyte 2 (POS 103). All POS data transfers through a multiplexed bus configuration on data lines 0 through 7 ( $D_{0-7}$ ). The user programs the ID bytes to his unique card ID value.

The Card-Enable register establishes the required card-enable function, and the POS 103 register is available for user-defined configuration data. The Micro Channel-bus input signals that determine the POS cycle are the Card Setup (CDSETUP), Lower Address lines ( $A2$ ,  $A1$ ,  $A0$ ), Memory or I/O Cycle (M/I/O) signal, and Bus Status ( $\overline{S1}$ ,  $\overline{S0}$ ) signals.

To meet Micro Channel bus-driving requirements, the circuit uses an external 74245 bus data transceiver. This arrangement isolates the local data bus from the Micro Channel data bus. With a 15-ns address-decoding device, the 30-ns  $T_{PD}$  of the 5AC324 easily meets the timing requirements for the CDSFDBK and POS signals. Because timing requirements are fairly tight, an extended cycle would need faster circuitry outside the 5AC324.

A designer can easily modify this 8-bit interface into a 16-bit interface by adding an additional transceiver, Card Data Size 16 (CDDSI6), and controls for it. By utilizing the unused outputs and decreasing the POS logic as needed, the design could be expanded to control memory transactions. □



**AN INTERFACE** for an 8-bit I/O slave to a PS/2 Micro Channel bus using an 5AC324 EPLD can save board space and power.



# SIMPLIFY A RISC EMBEDDED- CONTROLLER INTERFACE USING A PLD

A PLD IMPLEMENTS  
THE BURST LOGIC,  
TIMING CONTROL,  
TIMING ANALYSIS,  
LOGIC DESIGN, AND  
PROGRAMMABLE LOGIC.

THOM BOWNS

Intel Corp., 1900 Prairie City Rd., Folsom, CA 95630;  
(916) 351-8080.

© Intel Corporation, 1990

Reprinted from *Electronic Design*, January 25, 1990, A Penton Publication.

Designers are turning to reduced-instruction-set computer (RISC) processors, such as the 80960KB, for embedded-control systems because they deliver the necessary speed and streamlined operation. Design problems arise, however, when RISC processors are used. For example, RISC processors have more complex interface needs than do slower, conventional embedded controllers.

With the 80960KB, designers can take advantage of the 85C960 programmable-logic device, which implements the difficult portions of the interface and easily configures the programmable logic. The difficult portions that are handled by the application-specific PLD include the burst logic and timing control with all of their state tables, timing analysis, and logic design. The programmable logic implements flexible address decode and complete wait-state coverage.

The 80960KB's burst-mode bus and high-performance RISC architecture make system designs efficient in terms of size and space. But linking the chip with peripheral devices, such as memory or I/O devices, can require a fair amount of circuitry. Some portions of the interface vary little from one application to another, while other parts are customized according to the application. Designing this type of interface can be difficult and time consuming.

The basic 80960KB interface circuitry is composed of several discrete sections, including address latching, address decode, data transceivers, burst bus-control logic, wait-state generation and Clock/Reset synchronization (Fig. 1). The application-independent sections—address latching, data buffering, Clock/Reset synchronization, and burst bus-control logic—require high-performance, high-integration logic. These portions won't necessarily be affected by peripheral changes or other customization and fine tuning of the system. The address-latch portion calls for four byte-wide flow-through latches and an inverter for ALE (Address Latch Enable).

The circuitry for the Clock/Reset synchronization consists of several discrete high-speed logic devices and should have sufficient drive to supply all of the necessary clocking and reset pulses. Burst logic decodes bus-cycle information and offers addressing, burst cycle status, and other information—to assist the rest of the interface in handling burst Read/Write accesses.

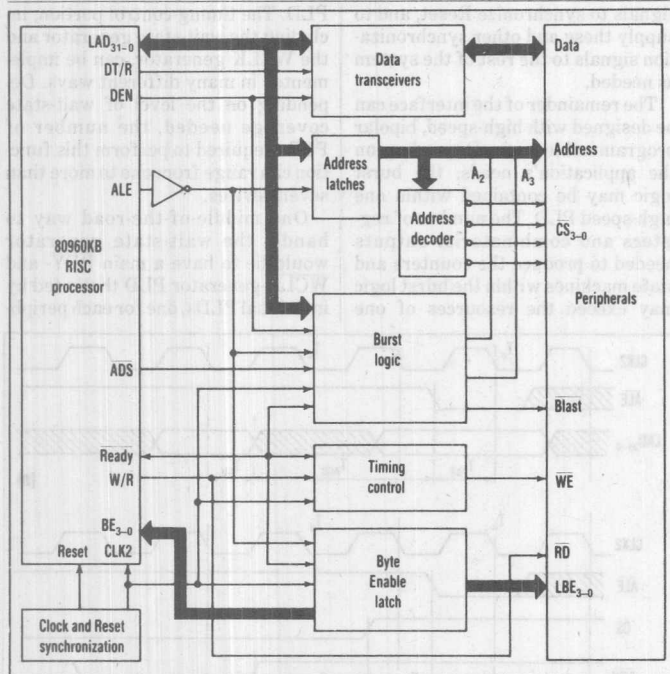
The remainder of the interface is application-specific. This portion includes Address Decode (or Chip Select Decode), wait-state generation, and optional circuitry to create such signals as WCLK (Write Clock) and Blast (Burst Last), which is used by burst-mode memories. The

## PERIPHERAL WAIT STATES

	First read	Subsequent reads	First write	Subsequent writes
Flash	2	2	5	5
Burst EPROM	1	0	—	—
SRAM	1	1	1	1
I/O	3	3	4	4

ELECTRONIC DESIGN  
JANUARY 25, 1990

## DESIGN APPLICATIONS PLD-BASED RISC INTERFACE



**1. THE BASIC 80960KB RISC-PROCESSOR** interface consists of application-independent and application-dependent sections. Portions that remain the same regardless of the application include address latching, data buffering, Clock/Reset synchronization, and burst bus-control logic.

address decoder, which is programmed to fit designers' memory maps, supplies latched Chip Selects to the various peripheral circuits. The wait-state generator reads the Chip Selects to determine which wait state count value to use, and then looks for a "go" signal from the bus state tracker to begin counting out wait states. Blast and WCLK are generated by way of state machines that work closely with the burst-control logic.

### TRADE-OFFS AND PITFALLS

Designers must balance several trade-offs affecting the interface circuitry that ties the 80960KB RISC processor to its peripheral devices. The address decoder, for example,

must be flexible enough to fit designers' memory maps, and it must be fast enough to select the peripherals without adding to the peripheral access time. If the decoder is placed after the address latches, the timing path would include a clock to ALE ( $T_{CO}$ ), latch delay, decode delay, peripheral-access time, data-transceiver delay, and setup time to the sampling edge in the data state ( $T_{CO}$ ) (Fig. 2a). The cumulative delay of  $T_{ACC}$  equals four CLK2 periods minus  $T_{CO}$  and  $T_{SET}$ . In a 20-MHz system with zero wait states, this would mean a period of 100 ns minus 23 ns, leaving no more than 77 ns for latch, decode, peripheral access, and transceiver delays.

The wait-state generator also in-

volves some trade-offs: compactness and complexity. The wait-state generator should cover all of the possible wait-state requirements for the various peripherals, which include differentiating between read and write accesses, and first and subsequent accesses within a burst transfer. The amount of logic needed to supply comprehensive wait-state coverage depends on the application. A single-PLD counter alone can handle only a few wait-state count values. Another approach would be to have individual PLD wait-state counters associated with each peripheral. Each wait-state counter's output would feed a main ready generator (Fig. 2b).

Nevertheless, only three clock edges are encountered from address time to the end of a data or wait state. The circuitry must decode peripheral selects, load in the correct wait-state count value, count down, and return a ready/not ready indication before the setup to the sampling clock edge (Fig. 2b).

Generation of the Blast signal for burst memories is rather tricky because the signal must occur during the data state of the last transfer. The circuitry controlling this signal must read the burst-progress information from the burst-logic section, and then assert Blast during the data state. The only way to determine whether the current state will be a data state or a wait state is to look ahead at the ready generator and anticipate its assertion by one CLK2 cycle (Fig. 2c).

WCLK must be qualified with the Latched Byte Enables to write to individual bytes of static RAM (SRAM). Because this will delay WCLK to some degree, it's necessary to delay  $A_2$  and  $A_3$  relative to WCLK to achieve a correct timing relationship between those signals (Fig. 2d).

The heart of the interface is the burst-logic section. It consists of several counters and state machines, which include the burst-size counter,

## DESIGN APPLICATIONS

# PLD-BASED RISC INTERFACE

the address counter ( $A_2$  and  $A_3$ ), and the Blast generator. The burst logic must sense the beginning of a burst access, load the burst-size down-counter from local address lines  $LAD_0$  and  $LAD_1$ , and store the initial low-order address values that were read from  $LAD_2$  and  $LAD_3$  in the address counter.

As each transfer is completed—signaled by the assertion of  $RDY$ —the burst-logic circuitry increments the address counter, decrements the burst-size counter, and determines whether additional transfers remain in the burst access. If there are more burst transfers in the current burst cycle, the burst logic signals the wait-state generator and the other state machines to proceed with another transfer.

To further complicate matters, these counters and state machines must be made codependent on each other and the rest of the interface. Designers must carefully plan each element in the burst logic, keeping in mind the states and timings of signals from the other portions of the interface.

## PROGRAMMABLE LOGIC

Interface circuitry is typically implemented in fast discrete TTL devices and bipolar programmable-logic devices. When using TTL and programmable logic, most designers break the circuit into discrete gates whenever possible. The remainder of the circuit is described in terms of registered functions or state machines.

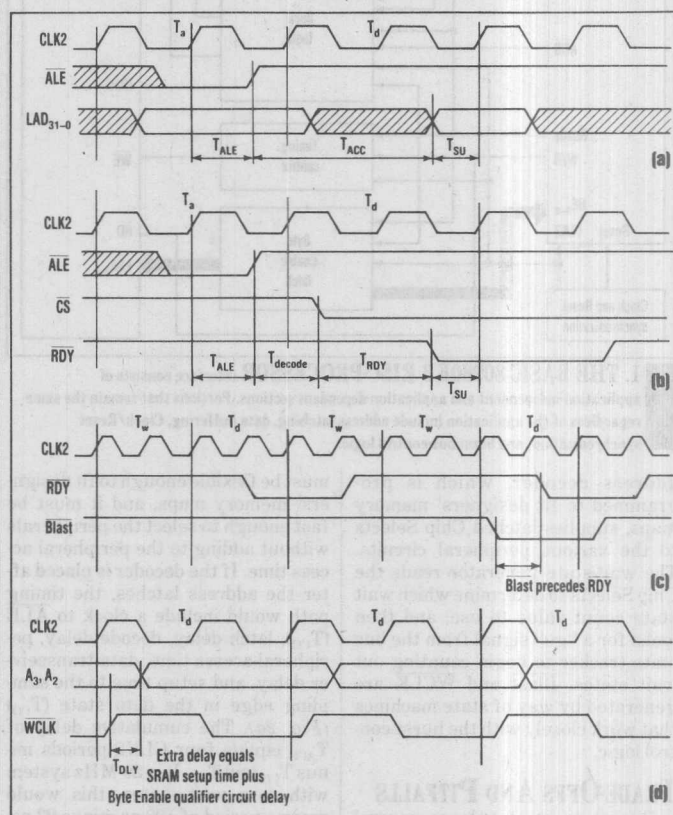
Most of the application-independent areas of the interface can be handled by high-speed TTL. The address latches and data transceivers would most likely be constructed of fast TTL devices, such as the 74F373 and 74F245, respectively, along with a 74F00 that would be used to invert  $ALE$ . A 74F138 could generate Chip Selects by decoding the latched high-order addresses. Several fast logic gates and registers could generate the full-speed and half-speed clock

signals to synchronize Reset, and to supply these and other synchronization signals to the rest of the system as needed.

The remainder of the interface can be designed with high-speed, bipolar programmable logic. Depending on the application's needs, the burst logic may be contained within one high-speed PLD. The number of registers and combinatorial outputs needed to produce the counters and state machines within the burst logic may exceed the resources of one

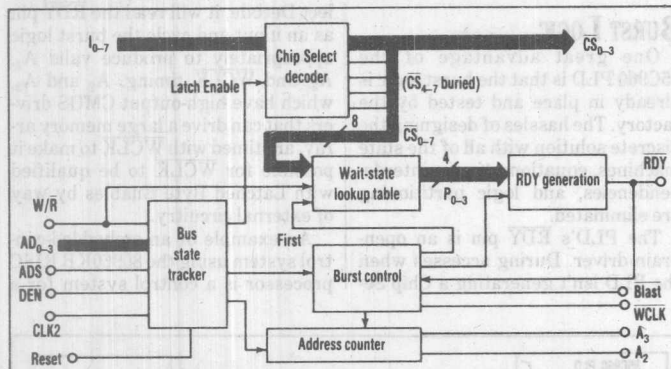
PLD. The timing-control portion, including the wait-state generator and the  $WCLK$  generator, can be implemented in many different ways. Depending on the level of wait-state coverage needed, the number of PLDs required to perform this function can range from one to more than seven devices.

One middle-of-the-road way to handle the wait-state generator would be to have a main  $RDY$ - and  $WCLK$ -generator PLD that's fed by individual PLDs, one for each periph-



**2. THE TIMING RELATIONSHIPS** between the interface signals are critical (a, b, c, d). For example, the interface circuit must decode peripheral selects, load in the correct wait-state count value, count down, and return a ready/not ready indication before the setup to the sampling clock edge.

## DESIGN APPLICATIONS PLD-BASED RISC INTERFACE



**3. BOTH HARD LOGIC** and programmable resources are offered in one package with Intel's 85C960 PLD. It's an application-specific part designed to perform many of the 80960KB processor's interface functions.

eral. As a peripheral Chip Select is generated, a corresponding PLD is activated and sends an appropriate wait-state count value to the main RDY generator programmable-array logic (PAL). That PAL then counts down from the value it received and asserts RDY at the end of the count. Each individual PLD could then distinguish between first and subsequent accesses within a burst, and between read and write accesses. With this information, the programmable-logic device could furnish a proper wait-state value.

The easiest way to create the burst logic is to determine the various elements—the burst-size counter and bus-state tracker, the address counter, and the burst-control logic—and describe each element in terms of state diagrams. The processor-bus states must be understood to properly design the bus-state tracker and relate it to the other state machines. The timing-control portion consists of individual programmable-logic devices that handle the various wait-state counts, and a main RDY down-counter PLD, which also creates WCLK.

This 80960KB-based design requires the WCLK generator to be described as a state machine, while the

RDY generator is just a pre-loadable down-counter with additional start, stop, and reset logic. By fully describing the burst logic and timing control as a set of linked state machines, designers can determine the amount of PLD resources needed. The state-machine descriptions are translated into PLD source language, either directly into state-machine or other high-level format, or as minimized Boolean equations.

### PARTITIONING

Designers then partition the design to fit into the available PLD resources and create simulation vectors to test the design's individual elements. The next step is to compile and simulate the resulting source files, which are programmed into the PLDs after debugging.

One major disadvantage of this solution is the address decode's inflexibility. The Chip Selects generated by the hard-logic device don't allow for easy restructuring of the memory map when additional memory is added or existing mapping is swapped. The inflexibility, combined with the tedium of designing the counters and state machines, partitioning all of that logic into the available PLD resources, and trying to end up with

a compact, low-power solution, creates lots of extra time and effort that could be spent improving the system in other areas.

Intel's 85C960 PLD is an advanced, CMOS, application-specific PLD that performs the major interface functions for 80960KA/KB-based systems. This PLD doesn't perform the entire 80960KA/KB interface, but it contains select decode, timing control, and burst logic in one 28-pin package. The 85C960 PLD was designed to deliver high-performance control for burst cycles without sacrificing configurability. Application-specific PLDs are the only programmable devices that offer high-performance, high-integration logic with a satisfactory degree of flexibility.

The chip contains an eight-input, eight-output (four external) address decoder (Fig. 3). Each output is independently configurable to respond to any address condition and is latched by an internally generated Latch Enable. In addition to having the flexibility of complete programmability, the address decoder generates the Chip Selects as fast as 10 ns after it receives valid addresses.

The PLD also has a programmable wait-state generator with a comprehensive-lookup table. There are four wait-state values associated with each Chip Select. When a given select is asserted, one of the four values associated with that Chip Select is loaded into the wait-state counter. The value depends on whether the current access is the first or second through fourth in a burst, and whether it's a Read or Write. This supplies the complexity to cover any wait-state requirement for each peripheral addressed by the PLD without the need for vast amounts of individual logic devices.

The burst logic is also on the chip. This incorporates the bus-state tracker, address cycling, and other timing-control logic, such as WCLK, and Blast. During a burst access, the bus-state tracker generates an inter-



## DESIGN APPLICATIONS

# PLD-BASED RISC INTERFACE

nal Address Latch Enable for the Chip Select decoder, and loads the wait-state generator with the appropriate count value. The tracker reads the low address inputs for burst size and starting address, then places  $A_2$  and  $A_3$  out on the address bus. During the burst, it keeps track of how many accesses remain in the burst, cycles  $A_2$  and  $A_3$ , strobes  $\overline{WCLK}$  appropriately, recycles the wait-state generator, and asserts  $\overline{Blast}$  while the last access is occurring.

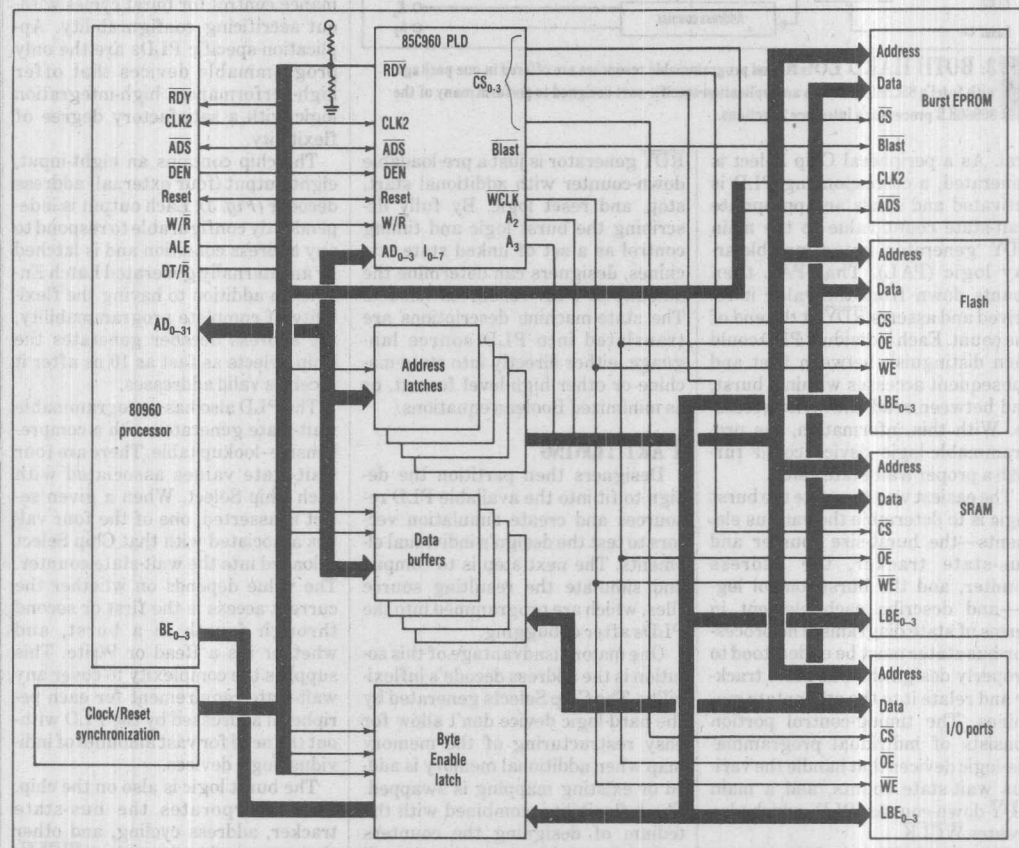
### BURST LOGIC

One great advantage of the 85C960 PLD is that the burst logic is already in place and tested by the factory. The hassles of designing the discrete solution with all of the state machines, equations, timing interdependencies, and logic partitioning are eliminated.

The PLD's  $\overline{RDY}$  pin is an open-drain driver. During accesses when the PLD isn't generating a Chip Se-

lect Decode, it will read the  $\overline{RDY}$  pin as an input and cycle the burst logic appropriately to produce valid  $A_2$ ,  $A_3$ , and  $\overline{WCLK}$  timing.  $A_2$  and  $A_3$ , which have high-output CMOS drivers that can drive a large memory array, are timed with  $\overline{WCLK}$  to make it possible for  $\overline{WCLK}$  to be qualified with Latched Byte Enables by way of external circuitry.

An example of an embedded-control system using the 80960KB RISC processor is a control system for a



**4. REMOTE WEATHER MONITORING** is performed by this embedded system built with the 80960KB processor, memory, and I/O devices. A PLD implements the interface between the processor and other elements.

## DESIGN APPLICATIONS

# PLD-BASED RISC INTERFACE

remote weather monitoring unit (Fig. 4). The unit might be placed in far-flung locations to monitor weather conditions, perform high-speed calculations, and then quickly return digested data to the main weather station.

The weather monitoring unit features a video-display-terminal interface, a serial microwave-transmitter link, and sensor interfaces. This type of system could operate continually in various climate conditions, and perhaps for long periods on backup power. As a result, the system would need highly reliable, low-power components. Because the unit is in a remote site, it would be checked for routine maintenance perhaps only twice a year. It should have some way of receiving new instruction code over the microwave link and storing the code in nonvolatile memory. If a hardware failure is detected, the single embedded-controller board could be removed and replaced in minutes.

The unit is built around the 80960KB embedded-control system with memory and I/O devices. Burst EPROM takes full advantage of the speed and width of the processor's burst bus. Flash memory is used for nonvolatile data storage and for updatable-code space. A certain amount of SRAM is needed by the processor for stack and scratch-pad memory. And with the I/O ports, the embedded controller can communicate with the various weather sensors, the video-terminal link, and the microwave transceiver.

## DESIGNING THE INTERFACE

The processor is interfaced to the four peripheral subsystems by way of the PLD and the other circuitry. The local address/data bus runs through a set of TTL transceivers to the system data bus, and through TTL latches to feed the system address bus. Byte Enable signals are latched in a high-speed PLD and routed to the peripherals that need them. The processor supplies the re-

mainder of the interface, which includes the address decode, wait-state generation, burst control, WCLK, and Blast generation.

The local (unlatched) address/data lines, LAD<sub>31-24</sub> and LAD<sub>3-0</sub>, are brought into the PLD along with the processor's bus-control signals ADS, DEN, CLK2 (high-speed clock), Reset, and W/R. RDY is terminated with a pull-up/pull-down network, and feeds the processor's Ready input as well as the Byte Enable latch. To facilitate burst control, non-burst peripherals receive A<sub>2</sub> and A<sub>3</sub> from the PLD rather than from the address latches. W/R from the processor ties directly to the Output Enable inputs for all of the peripherals, and WCLK from the PLD feeds the Write Enable lines for the peripherals that require a Write clock. Blast runs from the PLD to the burst-EPROM bank, along with ADS and CLK2.

According to the system-memory map, the burst EPROM is located at the bottom of memory for boot up (Fig. 5). Flash memory is located higher up, followed by SRAM. To allow for expansion, a large open space follows SRAM. The I/O ports are located at the very top locations of memory.

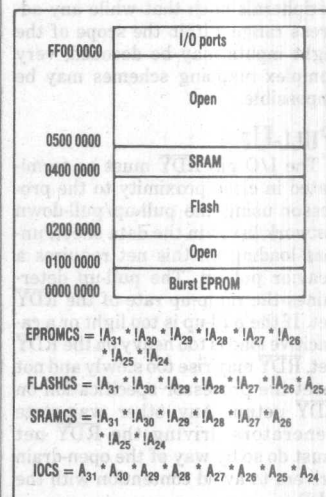
The wait-state values for all of the peripherals vary under the different burst-transfer conditions (see the table). Note that burst EPROM needs one wait state for initial reads, but needs no wait states for successive reads. Flash memory, with a 135-ns access time, requires two wait states for either reads or writes. Writing to the flash memories is accomplished by performing a command-data algorithm of writes and reads.

## DESIGN CONSIDERATIONS

There are several important items to consider when designing with the 85C960 PLD. Standard CMOS-design practices should be observed, such as a healthy bypass capacitor of at least 0.2 µF placed as close as possible to the V<sub>CC</sub> pin. Any unused in-

puts should be tied low to avoid high-power consumption. During normal operation, V<sub>CC</sub> must be ramped up to full voltage before any inputs begin to toggle, otherwise the device may enter an unknown state.

The timing of Reset versus CLK2 is critical. The 85C960 PLD relies on an internally generated clock signal which should be in phase with the bus clock. The first rising edge of CLK2 after Reset falls is considered by the PLD to be the "a" edge—the



### 5. THE SYSTEM memory map

illustrates that the burst EPROM is located at the bottom of the memory for system boot up. The equations are derived from the address locations shown.

PLD will time all of the inputs and outputs with this reference. Reset timing is consistent with the timing specification for the 80960KB processor.

Though the 85C960 PLD decodes eight Chip Selects, only four of them are routed externally. This might be troublesome for systems which have five or more peripherals. The easiest way to work around this is through the use of an additional, external ad-

## DESIGN APPLICATIONS

# PLD-BASED RISC INTERFACE

dress decoder, such as the Intel 85C508 programmable-address decoder. The external decoder could be programmed to supply Chip Selects in the ranges that shadow those given by one or more of the PLD's buried Chip Selects. This would make it possible for the PLD to continue to supply bus control for those peripherals as well. System designers should take note that the PLD Chip Selects are decoded with one product term. This affects the mapping of peripherals such that while any address range within the scope of the eight inputs may be decoded, very complex mapping schemes may be impossible.

### PULL-UP

The I/O pin  $\overline{\text{RDY}}$  must be terminated in close proximity to the processor using the pull-up/pull-down network listed in the data sheet, unless loading on this net requires a heavier pull-up. The pull-up determines the ramp-up rate of the  $\overline{\text{RDY}}$  net. If the pull-up is too light or a capacitive load is too heavy on the  $\overline{\text{RDY}}$  net,  $\overline{\text{RDY}}$  may rise too slowly and not meet the processor specification on  $\overline{\text{RDY}}$  setup. Any other wait-state generators driving the  $\overline{\text{RDY}}$  net must do so by way of the open-drain drivers to avoid contention with the PLD.

Although the burst logic is cycled appropriately when the 85C960 PLD isn't generating the Chip Select or  $\overline{\text{RDY}}$  signal,  $\overline{\text{Blast}}$  doesn't get produced in these circumstances. This is due to the timing relationship between  $\overline{\text{Blast}}$  and the sampled  $\overline{\text{RDY}}$  input (Fig. 2c, again).  $\overline{\text{Blast}}$  must be

asserted at the start of the last data state, and  $\overline{\text{RDY}}$  isn't necessarily valid until close to the end of the last data state. However, because the PLD will produce  $\overline{\text{Blast}}$  during accesses for which it decodes Chip Selects, designers merely need to ensure that the PLD decodes the Chip Selects for the burst memory.

The only portions of the 85C960 PLD that need configuration are the address decoder and the wait-state lookup table. The address decoder reads the eight address inputs, latches them, and decodes the address range according to the equation programmed in the product term of each individual Chip Select output. The Chip Selects then help determine lookup table will be loaded into the  $\overline{\text{RDY}}$  counter.

The Chip Select decode equations must be created first. In the example design, the Chip Select equations can be translated from the information given in the memory map (Fig. 5, again). Because eight address lines are brought into the Chip Select decoder, using the most significant eight address signals from the 80960KB processor allows for the widest range decode.

### DERIVING EQUATIONS

The equations are derived by taking the product of the true, or the complement, of the address lines that must decode a given block of memory. For instance, burst EPROM begins at address 00000000h. The most significant byte is 00, meaning that the upper addresses— $\text{LAD}_{31-24}$ —would be 00000000b. This produces the equa-

tion labeled EPROMCS. The memory block for flash memory requires twice the decode range than the burst EPROM block. Because the PLD is limited to one product term per output, the flash-memory block begins at a location following an open space after the burst-EPROM block. The equation decoding flash memory (FLASHCS) disregards the lowest of the incoming address lines.

The wait states for each peripheral are derived by determining the access-time requirements for each device and comparing that against the allowable time during each transfer state. Suppose, for example, that the allowable access time (including the decode, latch, and transceiver delays) for zero wait states is 55 ns. If the access time of the SRAM used in this design is 60 ns, at least one wait state would be required. An additional 50 ns are gained by the one wait state, making it possible for the use of up to 105-ns SRAMs.

The burst EPROM needs one wait state for address setup on the initial read access. However, subsequent reads during a burst transfer aren't required to setup addresses to the device because that's taken care of internally. Therefore, there are zero wait states in the subsequent burst transfers. The numbers for the rest of the peripherals are determined similarly. □

*Thom Bowns, technical support specialist for the programmable logic group at Intel Corp., has an AS in digital and microwave electronics from American River College, Carmichael, Calif.*

# Programmable AND/ Allocatable OR Based EPLD Addresses the Needs of Complex Combinational and Sequential Designs

by Todd K. Koelling  
Applications Engineer

3

## INTRODUCTION

Matching programmable logic applications with programmable logic devices has become a difficult task. Increasing demands for higher integration, higher performance and lower cost continue to drive system design engineers on to new technologies. The programmable logic industry has adeptly responded by supplying a wide variety of devices. At times, however, it is hard to differentiate these devices and to determine which makes the best solution for a particular application.

In a small way, this paper will attempt to differentiate devices and to determine which devices make the best solutions for groups of applications. This task will be accomplished by taking a general look at applications, the history of PLD arrays and a new device which solves several design problems.

## APPLICATIONS

College textbooks [1] on digital design teach that fundamentally there are only two types of applications: combinational and sequential. A combinational circuit generates outputs based on the immediate status of a group of inputs. A sequential circuit uses some mechanism to store data before generating the next set of outputs.

Inside combinational and sequential circuits are two fundamental elements: gates and registers. Gates are the prime component of combinational circuits where the output is an immediate function of the input. Registers are the static storage element added in sequential circuits to latch and hold data until the next cycle.

Figure 1 displays gates and registers graphically. The coordinates measure registers along the x-axis and gates along the y-axis. In this space, any combinational or sequential application can be displayed.

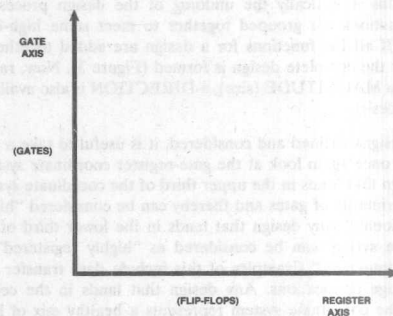


Figure 1. Gate/Register Coordinates

Common TTL functions are easily graphed. Figure 2 displays a comparator, storage register, shift register and counter. The comparator is a combinational circuit (purely gates) and hence lies along the gate axis. The storage register, on the other hand, is purely flip-flops and hence lies along the register axis. The shift register is primarily flip-flops—placing it close to the storage register—but it includes some gate logic, thus moving it up the gate axis. The counter is a good example of a function that lies somewhere in-between the two axes. The counter must store its current state, and thus leans heavily upon the registers, but it also uses a significant amount of gate logic to generate the next count state. The inclination toward the gate or register axis depends on the features the counter incorporates. Up and down count operation, clear and preload functions, and count enable/disable circuitry, all move the counter increasingly toward the gate axis. The magnitude of the counter (as with the other functions) depends



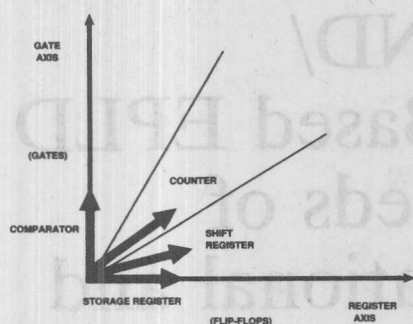


Figure 2. Common Functions on Coordinates

on the number of bit and features it includes. That is, a 16-bit counter is twice as large as an 8-bit counter which is twice as large as a 4-bit counter, provided the feature set remains the same.

Having examined functions, it is easy to examine complete designs. Each design can be decomposed into a group of function blocks. This is basically the undoing of the design process in which functions are grouped together to meet some high-level purpose. If all the functions for a design are added together, a vector for the complete design is formed (Figure 3). Now, rather than just a MAGNITUDE (size), a DIRECTION is also available for each design.

With designs defined and considered, it is useful to take a step back and once again look at the gate-register coordinate system. Any design that lands in the upper third of the coordinate system consists primarily of gates and thereby can be considered "highly combinational." Any design that lands in the lower third of the coordinate system can be considered as "highly registered" or "register intensive." Examples of this include data transfer and data storage applications. Any design that lands in the center third of the coordinate system represents a healthy mix of both gates and registers. This means it is probably a state machine or some sort of sequential application. Hence, the middle third region will be called the "state machine" region, though some state machines may land in the other two regions. The coordinate system with the three regions segmented and labelled is shown in Figure 4.

#### ARRAY ARCHITECTURES

Through the years, programmable logic devices have evolved by trying to meet the needs of combinational and sequential applications. This has been accomplished through higher integration, higher flexibility and higher performance and has resulted in the myriad of PLDs available today. Though the features have varied and expanded immensely, the core of the programmable logic device has remained virtually the same. It is this core—the implementation of the combinational logic array—which deserves a closer look.

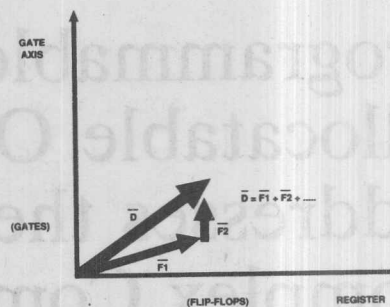


Figure 3. Design Vector

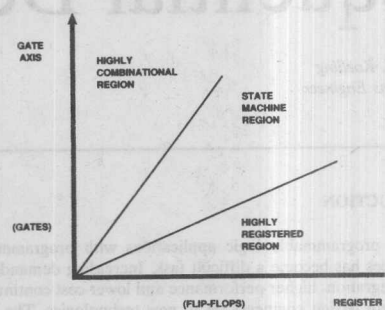


Figure 4. Application Regions

#### A Brief History of PLD Array Architectures

Programmable logic first appeared in 1975 with the introduction of the Field Programmable Logic Array (FPLA). With its programmable AND/programmable OR array, the FPLA was extraordinarily powerful for integrating combinational logic. Registers were later added to the FPLA outputs, creating the Field Programmable Logic Sequencer (FPLS). This device better attacked the needs of sequential applications.

In 1978, the FPLA was honed to a programmable AND/fixed OR array with the introduction of the PAL [2] device. Due to its ease of design and CAD tool support, the Boolean sum-of-products part quickly became the designer's choice. In addition, the PAL included registered versions with registered outputs and registered feedback. These two attributes made the devices ideal for state machines.

In essence, the last array architecture, fixed AND/programmable OR, has been around for many years in the form of the PROM and other PROM-based logic devices. It wasn't until 1984, however, that the SRAM-based LCA introduced the fixed AND/programmable OR array to the designer in an expedient form. Due to its large number of registers per device (122 or more), the LCA has provided an excellent programmable solution for register-intensive applications.

## Mapping of Array Architectures

Returning to the application graph developed earlier, each architecture and device is displayed (Figure 5). The combinational power and flexibility of the programmable AND/programmable OR architecture places the PLA and PLS devices in the highly combinational region.

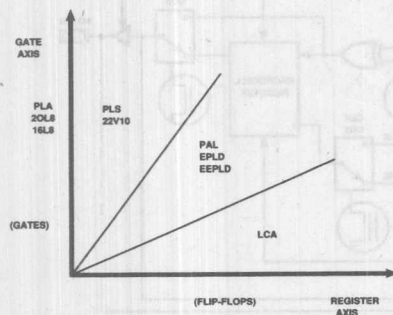


Figure 5. Device Application Areas

The registered output with registered feedback PALs are optimum for state machines while some PALs, such as the 22V10, are geared to provide a large amount of combinational logic per register. Thus, most of the registered PALs fit in the state machine region, while a few, like the 22V10, move up into the highly combinational region. (Logic PALs such as the 16L8 and 20L8 fit along the y-axis as they are purely gates.)

Architecturally, most EPLDs and EEPLDs have followed the track set by the PAL. They have added higher input and output flexibility, higher integration, and other worthwhile features, but the core of the architecture has remained the same—the programmable AND/fixed OR array. Thus the EPLD and EEPLD fit the same application regions as the PAL.

The highly registered region is best covered by the LCA where its high register count and fast toggle rates are well utilized.

Where will the next innovation be? The answer may not be in an innovation, but in a very practical advancement from Intel in 1988.

## THE PROGRAMMABLE AND/ALLOCATABLE OR ARRAY

Via a novel product term allocation [3] scheme, the best of the PLA and the PAL have been married into a single device. In the programmable AND/allocatable OR array architecture, each macrocell output can have anywhere from 0 to 16 AND terms allocated to its OR gate in increments of four. In this manner it is very similar to the PLA with its OR input flexibility, yet it retains the Boolean sum-of-products architecture used in the PAL which offers easy design entry and optimized CAD tools for minimization and fitting. The net result is a novel approach for efficiently addressing the needs of complex combinational as well as sequential applications.

## How P-Term Allocation Works

Figure 6 displays a macrocell of the 24-pin, 12 macrocell 5AC312. Each macrocell contains eight product terms grouped into two blocks of four. Each block has its own control switch that allows it to be retained by the macrocell or lent to a neighbor. Likewise, each macrocell can ignore or borrow a block from each of its neighbors. With two neighbors for each macrocell, each macrocell can have a total of 0, 4, 8, 12 or 16 p-terms allocated to its OR gate. This allows the device to shift resources toward those equations that require a high number of p-term inputs away from those that do not.

## Combinational Example: Micro Channel [4] Decoder

Figure 7 shows the 5AC312 pinout for the micro channel decode logic on a PS/2 [4] Ethernet [5] adapter. Based on the address lines, micro channel bus cycle signals, and POS/command register inputs, the 5AC312 generates the card select feedback (CDSFDBK#) and card data size 16 (CDDS16#) return signals for the micro channel. It also generates an asynchronous board select signal (ABDSEL#) that feeds an on-board arbiter. As memory is shared between the PS/2 motherboard and the adapter 82586 LAN coprocessor, the equations for all three of these outputs become quite complex (Figure 8).

Even after processing the equations through the industry-standard Espresso [6] minimizer employed by the Intel Logic Optimizing Compiler (LOC), the nested equations in Figure 8 expand out into the minimized equations in Figure 9. The board select, card select feedback, and data size 16 signals are 15, 14 and 14 product terms, respectively. This is not a problem for the 5AC312, however, as the LOC software automatically allocates the device's resources to four, four p-term blocks for each signal. The three equations use a total of 12 out of the 24 four p-term blocks available in the device—50% of its combinational capacity.

Though not used in this application, since the CDSFDBK#, CDDS16#, and ABDSEL# are all driven off unlatched address decodes, another feature which makes the 5AC312 attractive for address decoding and bus interfacing is its latched input capability. On the IBM PC/AT [7] bus, for example, the upper address lines (LA17-23), may need to be latched with the active edge of the bus address latch enable (BALE). Elsewhere in the micro channel interface, latching the address and status signals may be useful as both will go invalid about halfway through the command (CMD#) cycle. The 5AC312 is capable of latching up to eight inputs, each of which can be enabled individually or by the global latch enable.

The decode design could be implemented in a PLA or a 22V10, but the 5AC312 offers more strength over the PLA and more flexibility over the 22V10. Even with 32 and 48 p-terms feeding the programmable OR gates, most 24-pin PLAs would have trouble fitting these equations. With its fixed allocation architecture of 8, 10, 12, 14 and 16 p-terms, the 22V10 offers p-term resources comparable to those of the 5AC312 and can handle the equations. With configurable p-term allocation, however, the 5AC312 offers a more flexible solution. In cases where latching needs to be done, the address latching must be done external to the 22V10 since it does not have the ability to directly latch inputs. This impacts both board space and performance in a negative fashion.

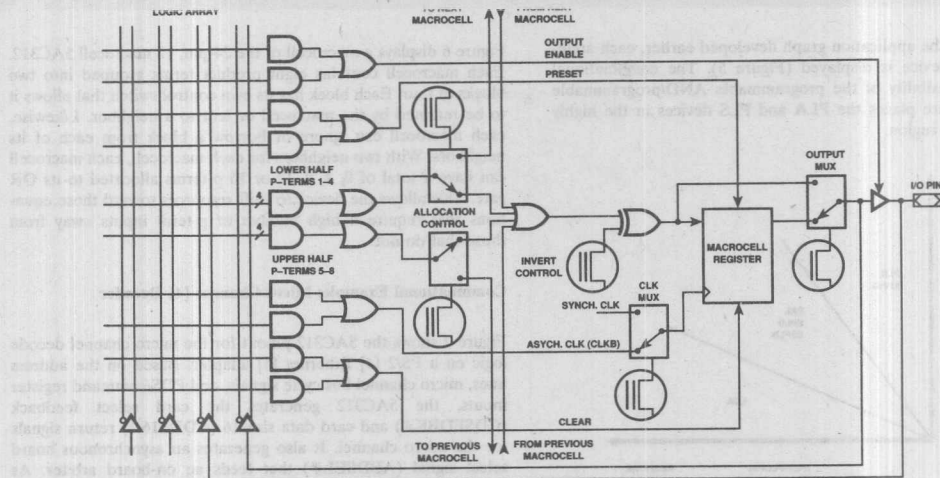


Figure 6. 5AC312 Macrocell Architecture

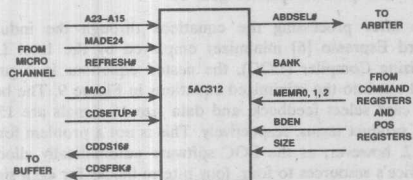


Figure 7. Pinout for Ethernet Adapter Decode Logic

#### Sequential Features

In addition to the allocatable OR architecture, the 5AC312 offers a host of other enhancements: separate register and pin feedback paths, register preset, and two product terms on clock, clear, preset and output enable control lines (Figure 6). These features, on top of the programmable AND/allocatable OR array, make the 5AC312 ideally suited for complex sequential designs.

Though not a state machine, the programmable baud rate generator circuitry for the Intel 8251 Universal Synchronous/Asynchronous Receiver/Transmitter does have a good mixture of gates and register functions, qualifying it for the state machine application region. The input baud rate (BAUDIN) is divided down to lower baud rates through a series of toggle flip-flops. Then, based on the select data stored in the D2, D1, D0 flip-flops, one of the divided-down baud rates is selected and sent out on the baud rate out pin (BAUDOUT).

Implementing the design in a standard 24-pin PLD (exemplified here by the Intel 5C060) is very costly. The data inputs must be latched inside a macrocell, using not only the macrocell but also the pin. The divide down toggle flip-flops cannot be buried, resulting in the loss of a pin for each flip-flop. The net utilization for the 5C060 implementation is 12 of 16 macrocells and 16 of 24 pins, virtually all of the device.

```

EQUATIONS:
AMB = A22 & A22 & A21 & A20 & MEMR2 & SIZE; % EXTENDED RANGE 32 K %
AMB = A23 & A23 & A22 & A21 & A20 & MEMR2 & SIZE; % EXTENDED RANGE 64 K %
AMB = A23 & A23 & A22 & A21 & A20 & MEMR2; % NOT EXTENDED RANGE %
AMB = A10 & A10 & A17 & A17 & SIZE; % EITHER, 32 K %
AMB = (A10 & MEMR1 & A10 & MEMR0) & #; % MATCH XC0000 %
AMB = (A10 & MEMR1 & A10 & MEMR0) & #; % MATCH XC0000 %
AMB = (A10 & MEMR1 & A10 & MEMR0) & #; % MATCH XC0000 %
AMB = (A10 & MEMR1 & A10 & MEMR0) & #; % MATCH XC0000 %
AMB = A10 & A10 & A17 & A17 & MEMR1 & A10 & MEMR0; % EXTENDED, 64 K %
AMB = (A10 & A17 & MEMR1 & A10 & MEMR0) & #; % MATCH F8000, 64 K %
AMB = (A10 & A17 & MEMR1 & A10 & MEMR0) & #; % MATCH F8000, 64 K %
AMB = (A10 & A17 & MEMR1 & A10 & MEMR0) & #; % MATCH F8000, 64 K %
AMB = A10 & A10 & A17 & A17 & SIZE & #; % NOT EXTENDED, 64 K %
AMB = (MEMR0 & A10 & #); % MATCH SC0000, 64 K %
AMB = (MEMR0 & A10 & #); % MATCH SC0000, 64 K %
AMB = MADE24 & REFRESH; % ALL CYCLES %
AMB = SETUP; % MEMORY CYCLES %
AMB = SETUP; % SETUP CYCLES %
CDS16 = (BDEN & IBANK & AM7 & AM5 & #); % ENABLED, SHAM, NOT SETUP AND %
CDS16 = (AM7 & AM4 & #); % EXTENDED, 32 OR %
CDS16 = (AM7 & AM4 & #); % NOT EXTENDED, 32 OR %
CDS16 = (AM7 & AM4 & #); % NOT EXTENDED, 64 OR %
CDS16 = (BDEN & AM7 & AM5 & #); % NOT EXTENDED, 64 %
CDSFBK = (BDEN & AM7 & AM5 & #); % ENABLED, NOT SETUP AND %
CDSFBK = (AM7 & AM4 & #); % EXTENDED, 32 OR %
CDSFBK = (AM7 & AM4 & #); % NOT EXTENDED, 32 OR %
CDSFBK = (AM7 & AM4 & #); % NOT EXTENDED, 64 OR %
CDSFBK = (AM7 & AM4 & #); % NOT EXTENDED, 64 %
ABDSEL = (AM1 & AM4 & AM7 & AM5 & #); % EXTENDED, 32 OR %
ABDSEL = (AM2 & AM5 & AM7 & AM5 & #); % NOT EXTENDED, 32 OR %
ABDSEL = (AM2 & AM5 & AM7 & AM5 & #); % EXTENDED, 64 OR %
ABDSEL = (AM2 & AM5 & AM7 & AM5 & #); % NOT EXTENDED, 64 OR %
ABDSEL = (AM7 & AM5 & #); % SETUP CYCLE %

```

Figure 8. Micro Channel Decoder Nested Input Equations

Implementing the same design in the Intel 5AC312 uses a much smaller amount of space. By using the input latches available on the 5AC312, the select data inputs can be stored immediately at the input pin rather than inside a macrocell. This saves a macrocell, saves a pin, and decreases the delay time. Second, since the 5AC312 has separate register and pin feedbacks on each macrocell, the baud rate divider can be buried by using the register feedback paths while the input feedback paths remain available for use as standard inputs. Inside the 5AC312, the circuit consumes 8 of 12 macrocells, and 7 of 24 pins, a significant I/O pin savings.

In fact, the I/O pin savings is so significant that the accompanying address decode circuitry—which would be implemented typically in a 20L8 or second PLD—can be added to the 5AC312

EQUATIONS:  
ABDSEI

[illegible][illegible]

```
* BDEB BANK MADEN4 REFRESH SETUP M JO A23 A22 *  
A21 A20 MEMR3 SIZE A19 A18 A17 A16 MEMR1 A15 MEMR0  
BDEB BANK MADEN4 REFRESH SETUP M JO A23 A22 *  
A21 A20 MEMR3 A19 A18 A17 A16 MEMR1 A15 MEMR1  
BDEB BANK MADEN4 REFRESH SETUP M JO A23 A22 A21  
A20 MEMR3 SIZE A19 A18 A17 A16 MEMR1 A15 MEMR1  
BDEB BANK MADEN4 REFRESH SETUP M JO A23 A22 A21  
A20 MEMR3 SIZE A19 A18 A17 A16 MEMR1 A15 MEMR0  
BDEB BANK MADEN4 REFRESH SETUP M JO A23 A22 A21  
A20 MEMR3 SIZE A19 A18 A17 A16 MEMR1 A15 MEMR1  
BDEB BANK MADEN4 REFRESH SETUP M JO A23 A22 A21  
A20 MEMR3 SIZE A19 A18 A17 A16 MEMR1 A15 MEMR0  
BDEB BANK MADEN4 REFRESH SETUP M JO A23 A22 A21  
A20 MEMR3 SIZE A19 A18 A17 A16 MEMR1 A15 MEMR1  
BDEB BANK MADEN4 REFRESH SETUP M JO A23 A22 A21  
A20 MEMR3 SIZE A19 A18 A17 A16 MEMR1 A15 MEMR1  
BDEB BANK MADEN4 REFRESH SETUP M JO A23 A22 A21  
A20 MEMR3 SIZE A19 A18 A17 A16 MEMR1 A15 MEMR1
```

(Figure 11). The 14 address inputs (A13-A0), along with the memory or I/O status signal (M/IO) are fed into the PLD to generate the baud rate select data clock signal (BAUDEL) and the 8251 Command/Data (C/D) and Chip Select (/CS) signals. The net result is a 10 of 12 macrocell, 24 of 24 pin, single-chip solution.

Based around a novel programmable AND/allocatable OR array structure, the Intel 5AC312 is uniquely suited to cover both highly combinational and complex sequential designs (Figure 12). The 5AC312 is made combinational powerful through a 0 - 16 product term allocation arrangement and sequentially powerful through separate register and pin feedback and other features. The 5AC312's latched input capability is an asset in both combinational and sequential applications. The net result is a combinational powerful, sequentially powerful 24-pin device.

## ACKNOWLEDGEMENTS

Special thanks to David Poisner of the Intel Datacomm Focus Group in Folsom, CA and Prof. J. Michael Dunlap and Robert A. Miller of Willamette University in Salem, OR for the use of their designs in this paper.

## REFERENCES

- [1] An Engineering Approach to Digital Design, William I. Fletcher, Prentice-Hall Inc., 1980, pp. 276, 280, 281. (Provided as an example.)
- [2] PAL is a registered trademark of Monolithic Memories Inc.
- [3] Product Term Allocation is an Intel Patent Pending.
- [4] Personal System/2 and Micro Channel are trademarks of International Business Machines Corp.
- [5] Ethernet is a trademark of Xerox Corp.
- [6] ESPRESSO is a copyright of the University of California at Berkeley.
- [7] IBM PC/AT is a registered trademark of International Business Machines Corp.

**Figure 9. Micro Channel Decoder Expanded and Reduced Equations**



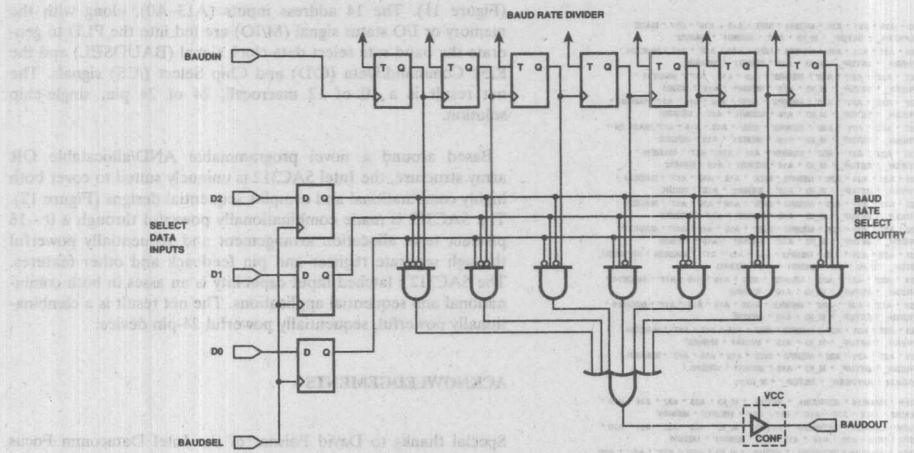


Figure 10. Programmable Baud Rate Generator Circuitry

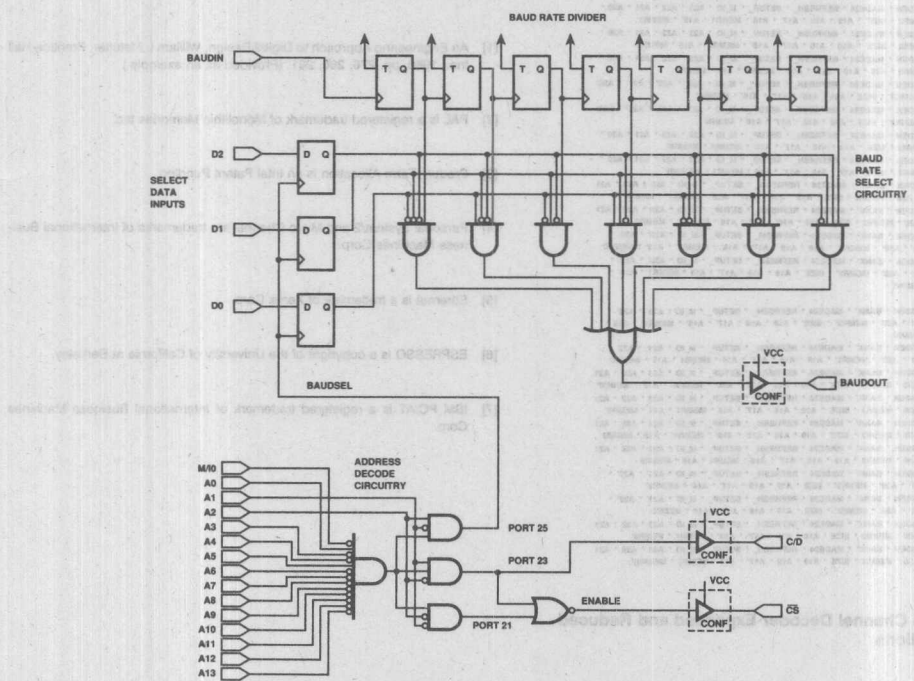


Figure 11. Programmable Baud Rate Generator

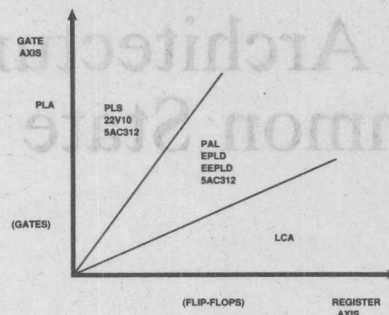


Figure 12. 5AC312 Application Areas

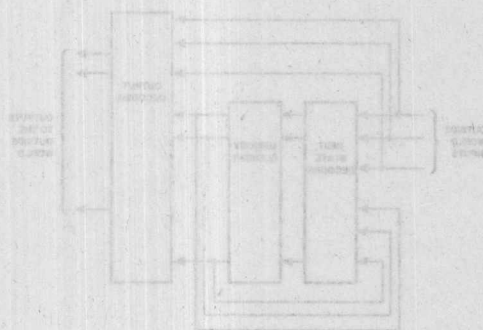


Figure 1. Class A State Machine

The Class B machine differs from Class A in that its output is only dependent on the present state of the machine through one or more combinational logic blocks.

$$z = f(y, s)$$

Class B Machine  
(Moore Type A Machine)

Finally, the Class C machine is similar to the Class B machine in that its output is only dependent on the present state of the machine through one or more combinational logic blocks.

$$z = f(y, s)$$

Class C Machine  
(Moore Type B Machine)

If a Class A or a Class B machine is implemented as a standard PLD, two characteristics would be required for each one: successful

The introduction of programmable logic devices (PLDs) as a new technology in the design world. It enabled engineers to integrate a variety of devices into a single device, thus saving space and power. This technology has been used in a variety of applications, such as in the design of control systems, data processing, and communication systems. The 5AC312 is a new PLD device that offers a variety of features, including a high density of logic cells, a high speed of operation, and a low power consumption. The 5AC312 is a new PLD device that offers a variety of features, including a high density of logic cells, a high speed of operation, and a low power consumption.

A typical programmable logic device is composed of a network of AND gates, OR gates, and flip-flops. The AND gates are used to combine the inputs of the device, and the OR gates are used to combine the outputs of the AND gates. The flip-flops are used to store the output of the device. The 5AC312 is a new PLD device that offers a variety of features, including a high density of logic cells, a high speed of operation, and a low power consumption.

### Types of State Machines Possible in PLD

The three basic types of state machines are Class A, Class B, and Class C. Class A is a Moore type machine, Class B is a Moore type machine, and Class C is a Moore type machine. The main characteristic of the Class A machine is that its output is only dependent on the present state of the machine. The main characteristic of the Class B machine is that its output is only dependent on the present state of the machine. The main characteristic of the Class C machine is that its output is only dependent on the present state of the machine.

$$z = f(y, s)$$

Class A Machine  
(Moore Type A Machine)

If a Class A or a Class B machine is implemented as a standard PLD, two characteristics would be required for each one: successful

# Advanced Architecture PLDs Solve Common State Machine Problems

by Liliyas S. Koumis  
Technical Marketing Engineer

## INTRODUCTION

The introduction of programmable logic devices (PLD) was a true revolution in the hardware design world. It enabled engineers to shrink circuits requiring several devices onto a single device thus simplifying their designs while saving space and power. Traditionally, PLDs have been used in combinational circuits such as address decoders as well as sequential circuits such as bus arbitration schemes. During the last few years, advances and improvements in PLD architectures enabled the devices to grow more complex while addressing the never-ending quest for higher density and faster speeds. Despite these improvements, engineers still face certain problems and limitations when implementing state machine designs with PLDs. The Intel 5AC312 and 5AC324, multi-purpose generic erasable PLDs, offer a solution to these problems that gives engineers a better device to implement their designs.

A typical programmable logic device is composed of a user-programmable AND array, a fixed OR gate, followed by an output register which includes a feedback path from the output to the programmable AND array. Combination of these elements is commonly referred to as a 'macrocell.' The existence of a feedback path from the output registers to the AND array makes PLDs ideal candidates for state machine implementations.

## TYPES OF STATE MACHINES POSSIBLE IN PLDs

The three basic categories of state machines are Class A, Class B and Class C, better known as MEALY, MOORE TYPE A and MOORE TYPE B respectively. It is possible to implement any of the classes of state machines in a PLD, however the efficiencies vary with state machine class. The main characteristic of the Class A machine is that its outputs to the external world are a function of both the input and the present state of the machine as shown in Figure 1. Mathematically, this can be expressed as:

$$z^{n+1} = f(x^n, y^n)$$

Class A Machine  
(Mealy Machine)

where 'z' is the decoded output, 'x' is the input, 'y' is output of the next state decoder and 'n' is the present state.

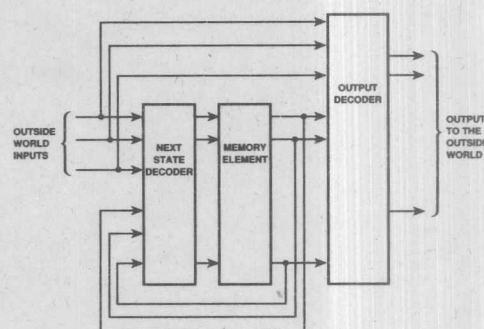


Figure 1. Class A State Machine

The Class B machine differs from Class A in that its output is only dependent on the present state of the machine through output decoding, or better expressed as:

$$z^{n+1} = f(y^n)$$

Class B Machine  
(Moore Type A Machine)

Finally, the Class C machine is essentially the same as the Class B but requires no output decoding: the outputs are the next state of the machine:

$$z^{n+1} = y^{n+1}$$

Class C Machine  
(Moore type B Machine)

If a Class A or a Class B machine is implemented in a standard PLD, two macrocells would be required per state; one macrocell

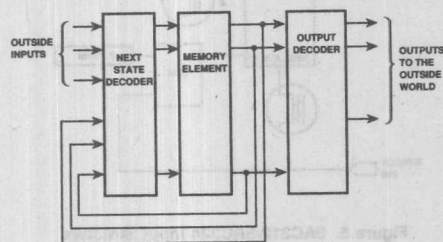


Figure 2. Class B State Machine

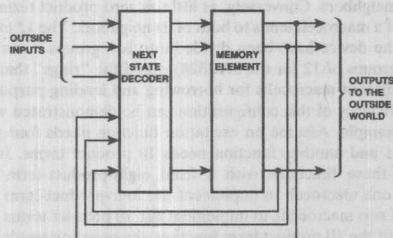


Figure 3. Class C Machine

has to be used for the input decoding and the other for the output decoding. The use of an extra macrocell for output decoding is not an efficient use of the device resources. For example, the largest state machine that can be implemented in an eight-register PLD is one with only four states variables. On the other hand, since a Class C machine does not perform output decoding, only one macrocell per state is required. As a result, that a machine with eight state variables could easily fit in the same eight-register PLD.

#### PROBLEMS WITH STATE MACHINE DESIGNS IN PLDs

Despite the architectural advantages of implementing a Class C machine, traditional PLDs still inherit serious timing problems and structural shortcomings.

The first problem is violation of setup and hold times of the output registers. This is encountered commonly in environments where the inputs are asynchronously changing with the device clock. Sources for these problems may be different data paths for each input or origination of these input signals from circuits that use a different clock than that used for the output registers. This kind of problem causes the output to glitch and may cause the machine to enter an invalid or incorrect state. The problem traditionally has been solved by adding external metastable-hardened registers to synchronize the inputs with the outputs. This solution,

however, has obvious drawbacks such as an increase in chip count, additional time delays, and an increase in power consumption. In some cases, an eight-register IC is added even when only a few inputs must be synchronized.

The second most common problem is missed input pulses of short duration. In modern and complex circuits, short pulses may arrive at the inputs and disappear at a faster rate than the PLD clock. This causes these inputs to be missed by the PLD, which in turn causes erroneous operation of the state machine. Traditional PLDs offer no solution to this problem. Engineers have had to resort to adding circuitry to ensure that the inputs are present long enough to be seen by the PLD clock. This additional circuitry further complicates designs, adds delays, increases power consumption and adds an unnecessary burden to the engineers.

Finally, the most frustrating problem for engineers is that the number of product terms available per macrocell is fixed. Engineers usually assign the states to the output pins randomly, write the equations and allow the software to determine whether the equations fit their chosen device. If the number of product terms required to implement the given equations is greater than the fixed number of available product terms, the designer devotes additional time and resorts to more 'innovative' approaches, such as breaking down the state machine into smaller parts to fit the device. This not only introduces additional time delays and reduces the effective use of the device, but it also increases development time and board cost.

#### SOLUTIONS TO THESE COMMON PROBLEMS

The above discussion illustrates the need for a more sophisticated generation of Programmable Logic Devices that address these problems. Two new Intel PLDs, the 5AC312 and 5AC324, are specifically targeted at fulfilling the requirements of better set-up and hold timing, faster input clock rate and flexible product terms. The AC in the part name stands for "Advanced CMOS," the 3 stands for third generation and 12/24 is the number of macrocells in each device. This family of Intel Erasable Programmable Logic Devices uses advanced CHMOS\* EPROM cells instead of polysilicon fuses as a logic control element. This process enhances the testability and reliability of the devices while significantly reducing power consumption. For the remaining portions of this paper, the 5AC312 will be referred for ease of reference, but for all practical purposes, the 5AC324 is functionally identical to the 5AC312 but contains twice the number of macrocells and ten register/latched inputs instead of eight.

As it can be seen in Figure 4, the 5AC312 has the architectural features of a Class C machine but also offers additional features address the issues discussed earlier. The input structure of the 5AC312 offers several programmable options, each addressing a particular need or problem.

To address the first problem-violation of setup and hold times of the output registers—the 5AC312 offers an additional register/latch input with a programmable clock. The clock can be the same as the output register clock shifted by 180°, a separate high frequency clock, or be generated by a product term from the logic array.

By using the first clock option, synchronization is achieved, and thus the risk of output glitches is minimized. By cascading the input and output registers and shifting the input clock 180° from the output register clock, an additional advantage is gained by



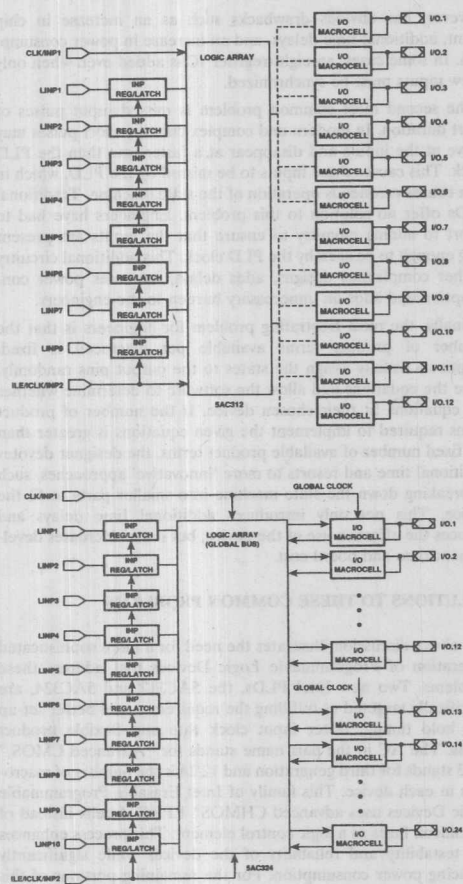


Figure 4. Architecturally Advanced 5AC312 and 5AC324 Global Block Diagrams

allowing enough time to satisfy the setup and hold time requirements of the output registers. Metastability characteristics of the device is of particular concern and are discussed later in this paper.

The second option, a separate high frequency clock, enables the device to sample inputs of very short time duration. This clock operates up to 50 MHz, with an input register setup of only 5 ns. If the latch feature is selected, the setup time is reduced to 0 ns. Of course, a mode can be selected where the input data flows through, bypassing the register/latch combination. The third clock option, clocking the input registers with the output of a product term from the logic array, is ideal for applications where registers are to be clocked only when a certain input condition is met.

To address the fixed product term problem, the 5AC312 implements an innovative solution called 'product term allocation.' In

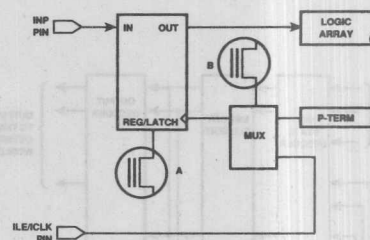


Figure 5. 5AC312/5AC324 Input Structure

each individual macrocell, the eight product terms are sub-divided into two groups of four. The product term allocation is achieved by allowing each of these four product term groups to be borrowed from or lent to adjacent macrocells. By 'allocating' product terms between adjacent macrocells, any register can be driven by as many as 16 product terms by borrowing unused product terms from its neighbors. Conversely, as little as zero product terms can be used if a macrocell lends to both of its neighbors. The 12 macrocells in the device have been divided into two groups of six each (or two groups of 12 for the 5AC324) called the "rings" that help define adjacent macrocells for borrowing and lending purposes.

The efficiency of this configuration can be demonstrated with a simple example. Assume an excitation function needs four product terms and another function needs 10 product terms. Implementing these functions with a fixed eight-product-term PLD requires one macrocell to implement the four-product-term function, and two macrocells to implement the 10 product terms function. To fit the 10 product term function, the equation needs to be broken into two parts, thereby increasing the delay. Therefore, out of 24 available product terms (three groups of eight), 14 are used ( $14/24 = 58\%$  efficiency). Using the 5AC312 to implement the same functions yields the following: the four-product-term function is implemented with half of a macrocell, allowing the other half to be allocated to the adjacent macrocell for implementing the 10-product-term function. No design-splitting is required. Therefore, out of 16 product terms, 14 are used. This translates to 88% product term utilization. The product term allocation is completely transparent to the user since it is achieved through software. When the compiler determines that an additional number of product terms is required, it automatically allocates resources to fit the required excitation function.

#### METASTABILITY CHARACTERISTICS

Although metastability is a relatively rare event, ignoring it can cause serious timing problems. The input registers found in the 5AC312 offer excellent recovery time where metastability is of concern. Metastability can be simply described as the inability of a register to decide the state of its output within a fixed amount of time. This event usually occurs when synchronizing an external event with a periodic clock. If a flip flop is clocked nearly at the same time as changing data, there is a small window of time where the output of the register is unknown. This window of time is the recovery time ( $t_{re}$ ) of the flip-flop and is typically in the order of nano-seconds. Designers at Intel have performed tests to obtain the recovery time for the 5AC3xx family of devices and have concluded that the Intel devices have better recovery times

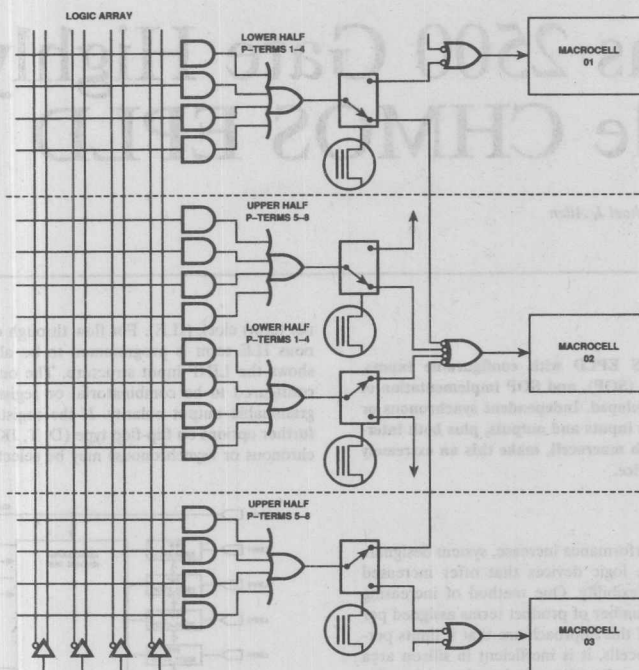


Figure 6. Product Term Allocation

than familiar TTL devices such as 74F74. Table 1 shows sample data taken at a clock frequency of 2 MHz and data frequency of 1 MHz.

Additional information on the procedure used to obtain this data and its use can be found in references one and three.

### CONCLUSION

Because of their internal architectural characteristics Programmable Logic Devices have become the ideal method to implement state machine designs. This is evident by the wide variety of applications where programmable logic devices are found today. The latest generation of PLDs, with the advantages of programmable I/O pins and expanded number affixed product terms, are certain

to replace traditional off-the-shelf logic as designers discover their usefulness in modern applications. However, to overcome the problems associated with setup and hold time violations, missed inputs and fixed number of product terms, a new generation of PLDs was needed. The Intel 5AC312 and 5AC324 overcome these problems by providing selectable input register/latch option with excellent metastability characteristics and allocatable product terms.

\*CHMOS is a patented process of Intel Corporation.

### REFERENCES

- [1] Chaney, Thomas J., "Measured Flip-Flop Responses to Marginal Triggering," IEEE Transaction on Computers, Vol. C-32 No. 12, December 1983.
- [2] Fletcher, William I., "An Engineering Approach to Digital Design," Prentice-Hall Inc., 1980.
- [3] Stoll, Peter A., "How to Avoid Synchronization Problems," VLSI Design, November/December 1982.
- [4] Weigl, Karl H., "Eliminating Common Problems in State Machine Designs Using Innovative PLD Architectures," SOUTHCON, December 1987.

Table 1

Device	Recovery Time (ns)
7474	1.6
74LS74	1.5
74S374	0.91
74F373	0.70
74F74	0.40
5AC312/5AC324	0.35

# A 15 ns 2500 Gate Highly Flexible CHMOS EPLD

by Ronald W. Swartz and Michael J. Allen

## ABSTRACT

A 2500 gate, 15 ns CMOS EPLD with configurable inputs, expandable sum of products (SOP), and SOP implementation of control signals has been developed. Independent synchronous or asynchronous clocking of the inputs and outputs, plus both internal and pad feedback of each macrocell, make this an extremely configurable 40-pin logic device.

## INTRODUCTION

As system complexity and performance increase, system designers are requiring programmable logic devices that offer increased density, performance, and flexibility. One method of increasing the flexibility is to vary the number of product terms assigned per output [1]. The drawbacks of this approach are that it limits performance of the large macrocells, it is inefficient in silicon area and internal chip resources, and it also restricts the options for the user pinout. By developing a 40-pin CMOS EPLD which offers configurable input structures, expandable SOP macrocells, SOP internal control signals, configurable output structures, and dual feedback paths, the demand for a high density, high performance, highly flexible device is met. Typical Tpd is 15 ns and Tco is 12 ns. The 1 $\mu$  N-well CMOS EPROM based technology allows for reconfigurability, low power operation, and 100% factory testing.

## CHIP ARCHITECTURE

The device consists of 24 I/Os, 10 Latched Inputs (LINPs), and 2 Clock/Inp pins, all of which are TTL compatible. In addition to these signals, internal feedback from each macrocell is also available as an array input, allowing the I/O pad to be used as an additional device input. Each I/O and LIN is configured on power-up by clocking the programmed values of EPROM cells through a shift register. There are 17 product terms (bit lines) per macrocell. These consist of 2 groups of 4 product terms for sum of product output, 4 sets of 2 product terms for sum of product control signals, and 1 bit line for the architecture bits of that macrocell. Twelve more product terms are required for asynchronous input clocking, input architectures, and the security bit (verify inhibit). Thus there are 120 word lines and 420 bit lines in the device. The 420 bit lines are divided into 4 quadrants with 6 macrocells each. The word lines are global to all 4 quadrants. Figure 1 shows a block diagram of the device.

The LINPs may be individually configured as either latches or registers, with either a synchronous (pin) or asynchronous (prod-

uct term) clock (ILE). For flow through operation, the asynchronous ILE term is programmed to be always enabled. Figure 2 shows the LINP input structure. The output macrocells may be configured to be combinational or registered, and support programmable output polarity. If the registered option is selected, further options on flip-flop type (D, T, JK, RS) and clocking (synchronous or asynchronous) may be selected. Each macrocell can

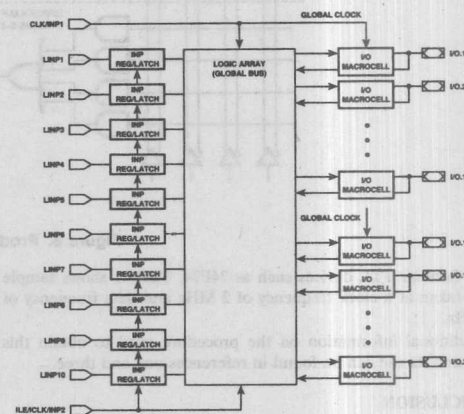


Figure 1. Block Diagram

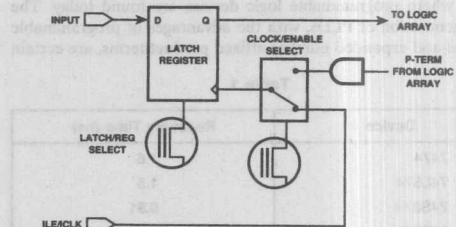


Figure 2. LINP Input Structure

Thus the number of product terms summed into a given output can range from 0 to 16. A macrocell with 0 data product terms retains its control product terms, and can still be used as an SR or T flip flop. Two product terms are summed together for each of the macrocell's control signals; OE, SET, RESET, and ASYNCH CLOCK. Figure 3 shows the output macrocell architecture.

### CIRCUITS AND PERFORMANCE

The high speed performance of the device may be attributed to a  $1\mu$  poly-silicide process, layout architecture, and circuit techniques. The process particulars will be described later in the paper, but were frozen for this design. Minimizing the parasitic delays was a high priority, and influenced several early layout choices. The large number of I/O macrocells presented a potential noise problem during multiple switching. To alleviate this, the device makes use of 2 Vcc and 2 VSS pins, which are located close to the center of the package, as shown in the pinout of Figure 4. This location has reduced pin inductance compared to the traditional corner placement, and significantly reduces power supply noise during simultaneous transitions.

The array of the device was broken into 4 quadrants of 105 columns each to reduce the rowline RC delays to less than 1 ns. All 4 quadrants are driven centrally from a single large row driver located at the end of the macrocell. By locating the row drivers here, the severe pitch limitations encountered by placing the row drivers on the end of the array are eliminated, and larger, faster tri-stateable drivers can be used. These drivers use high voltage p-channel devices to isolate the row drivers from the row lines during programming, with minimal speed degradation during normal mode. The bit line sense-amp, shown in Figure 5, contains

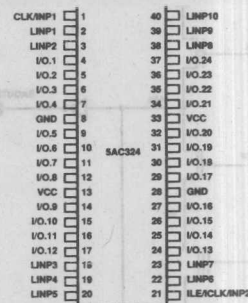


Figure 4. Device Pinout

unique biasing circuitry which limits the bit line swing to less than 150 mV, even when multiple cells are conducting. The amplifier uses a low threshold enhancement device to provide a pullup current, and a dynamic reference bias along with a current limiting device to limit the bit line swing. Response is very fast, even for recovery from super zero conditions, and typically takes less than 3 ns from row line input to sense amp output, allowing a Tpd of less than 15 ns. Figure 6 shows an oscillograph of typical Tpd data. Active Icc is less than 150 mA at 40 MHz. An option to trade off speed for power savings is also available to the user, which allows the part to operate in battery powered applications [2].

3

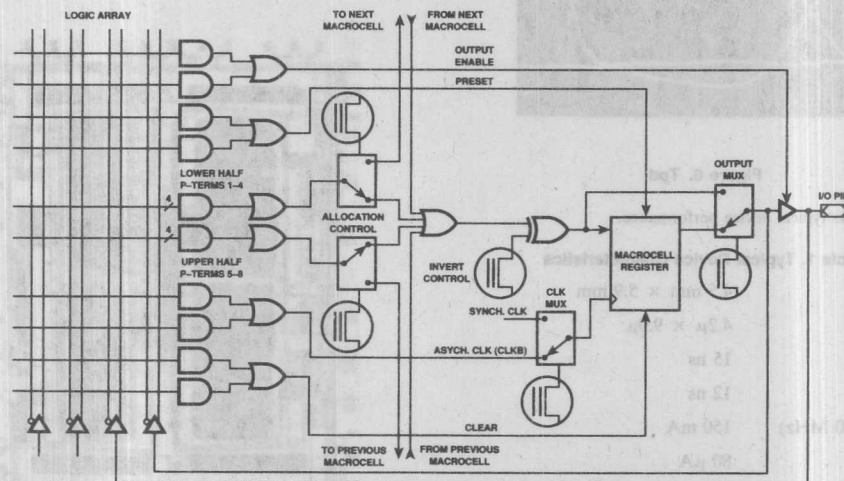


Figure 3. Output Macrocell Architecture



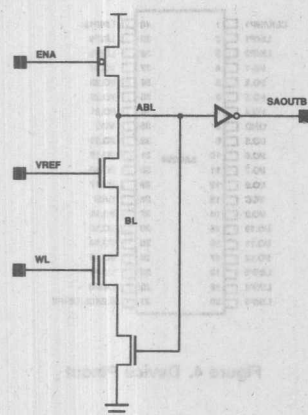


Figure 5. Bit Line Sense Amp

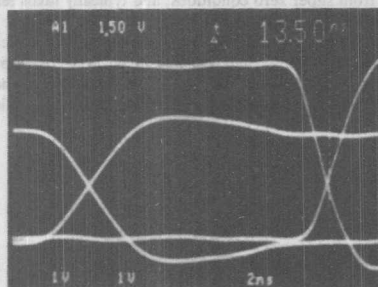


Figure 6. Tpd

Table 1 shows typical device performance.

Table 1. Typical Device Characteristics

Die Size	4.5 mm × 5.9 mm
Cell Size	4.2μ × 9.0μ
Tpd	15 ns
Tco	12 ns
Active Icc (40 MHz)	150 mA
Standby Icc	80 μA
Package	40-pin dip 44-pin PLCC
Technology	N-well 1.0μ CMOS EPROM with silicide

## PROCESS PARAMETERS

The device has been fabricated in a 1.0μ N-well CMOS EPROM technology, and utilizes 2 polysilicon and 1 metalization layers. Tungsten silicide is used to reduce poly interconnect resistance. Table 2 summarizes Process Parameters. Programming circuitry is handled by high voltage p-channel devices. The cell is a special double poly FLOTOX structure that has been optimized for PLD applications.

Table 2. Process Parameters

Technology	1.0μ CMOS EPROM Double poly with silicide
N-channel Leff	.9μ
P-channel Leff	.9μ
Tox	250Å
Polycide resistivity	4 ohms/sq
Polysilicon pitch	2.5μ
Metal-1 pitch	2.8μ
Contact cut	1.2μ × 1.2μ

## CONCLUSION

The combination of high speed, high density and flexible architecture makes this device an ideal solution for high speed microcomputer system design [3].

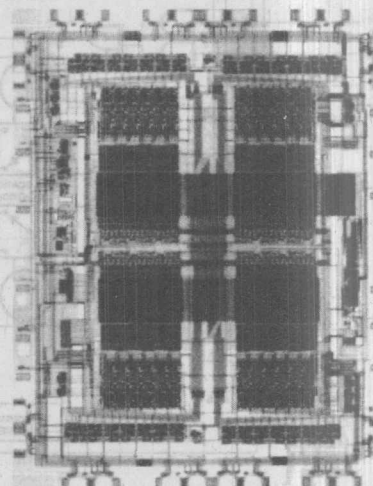


Figure 7. Die Photo

#### ACKNOWLEDGEMENTS

We would like to thank Jim Negrey and Reo Gargovich for their layout work, Chris Wawro, K.K. Ramakrishnan and Abid Asghar for their many helpful suggestions, and the Intel Fab personnel for producing the silicon.

#### REFERENCES

- [1] Advanced Micro Devices, "Programmable Array Logic Handbook," 1986.
- [2] S.C. Wong, et al, "CMOS Erasable Programmable Logic Device with Zero Standby Power," ISSCC Digest of Technical Papers, Feb. 1986, pp 242-243.
- [3] Intel Corporation, "Microcomputer Programmable Logic Handbook," 1989.

We would like to thank the many Intel Corporation customers who have helped us develop this document. We would also like to thank the many Intel Corporation customers who have helped us develop this document. We would also like to thank the many Intel Corporation customers who have helped us develop this document.

October 1989

# Implementing Cascaded Logic in EPLDs

**THOM BOWNS**  
**J. R. DONNELL**  
APPLICATIONS ENGINEER  
PROGRAMMABLE LOGIC

Order Number: 292003-002

## IMPLEMENTING CASCADED LOGIC IN THE 5C121

## CONTENTS

PAGE

**PROBLEM** ..... 3-136

**SOLUTION** ..... 3-136



## PROBLEM

Designs that utilize numerous levels of cascaded logic often result in excessive product terms when expressed in the sum-of-products form. Although this poses no problem when designing with discrete logic, EPLDs are generally optimized for the sum-of-product form. This stems from the architecture of the basic Macrocell.

Macrocells typically consist of a programmable AND array feeding a fixed width OR gate. Most Intel EPLDs (with a few exceptions) have a fixed OR gate width of eight product terms. For most applications, eight available product terms are sufficient. However certain designs require logic to be cascaded, which usually causes product term requirements to expand geometrically. One example where product terms become an issue is cascaded exclusive-OR (XOR) circuits. Here the number of product terms increase by 2 to the  $n$ th power, where  $n$  equals the number of XOR gates. If the number of product terms exceeds eight, the equation may not fit in the EPLD macrocell.

## SOLUTION

There is a simple solution to reduce the product term requirements when using cascading XOR (or other)

logic. Figure 1 shows a circuit cascading five exclusive ORs. As designed, this circuit expands to 32 product terms when expressed in the minimized sum-of-products form. (This is assuming that signals A thru F are single product terms themselves.) Figure 4 shows the minimized logic equation file produced by Intel's Logic Optimizing Compiler (iLOC).

An easy solution to fitting this logic into an EPLD is to cascade three exclusive ORs together and then send the result through some type of combinatorial feedback primitive, such as a COIF (Combinatorial Output—Input Feedback) or NOCF (No Output—Combinatorial Feedback). This signal can now be cascaded through two more XOR's to get the five total. This circuit is shown in Figure 2. Figure 4 shows the logic equation file for this implementation. Note the reduction in product terms from Figure 3.

The only penalty in this method is the added delay needed for the feedback path. The worst case  $t_{pd}$  (input to output delay) for the circuit in Figure 2 would be twice the specified  $T_{pd}$  (input to output delay) for the target EPLD. For the 10 ns 85C220, the  $T_{pd}$  would be 20 ns worst case as implemented in Figure 3.

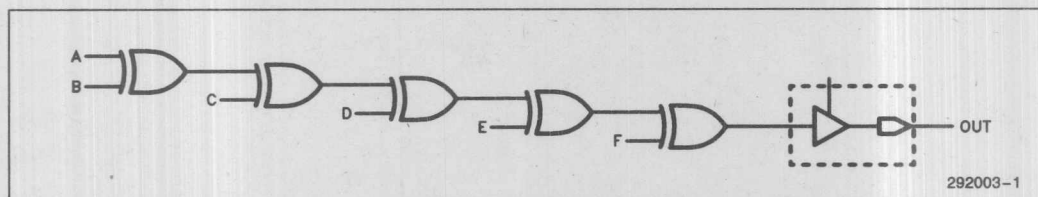


Figure 1. Cascaded Exclusive-ORs

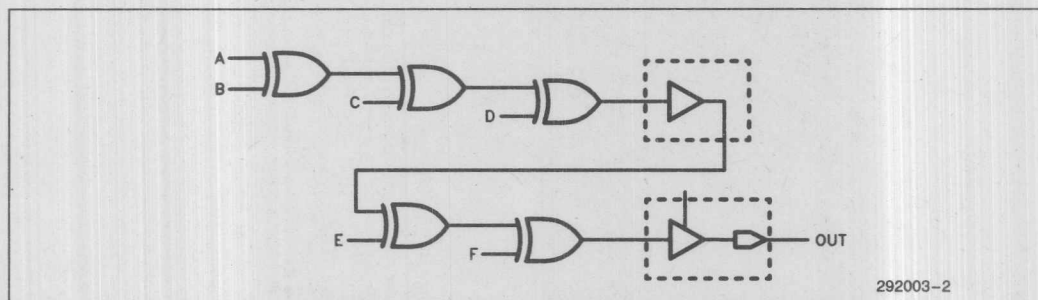


Figure 2. Cascaded Exclusive-ORs using Combinatorial Feedback

Cascading XOR gates--no feedback  
LEF Version 2.0 Baseline 4.3i  
22 Jul 1988

PART: 85C220

INPUTS: A, B, C, D, E, F  
OUTPUTS: O

NETWORK:

A = INP(A)  
B = INP(B)  
C = INP(C)  
D = INP(D)  
E = INP(E)  
F = INP(F)  
Q = CONF(OC, VCC)

EQUATIONS:

OC = F \* E' \* D' \* C' \* A' \* B'  
+ F' \* E \* D' \* C' \* A' \* B'  
+ F' \* E' \* D \* C' \* A' \* B'  
+ F \* E \* D \* C' \* A' \* B'  
+ F' \* E' \* D' \* C \* A' \* B'  
+ F \* E' \* D \* C' \* A' \* B'  
+ F' \* E \* D \* C \* A' \* B'  
+ F' \* E' \* D' \* C' \* A \* B'  
+ F \* E \* D' \* C' \* A \* B'  
+ F' \* E' \* D \* C' \* A \* B'  
+ F' \* E \* D \* C' \* A \* B'  
+ F \* E' \* D' \* C \* A \* B'  
+ F' \* E \* D' \* C \* A \* B'  
+ F' \* E' \* D \* C \* A \* B'  
+ F \* E' \* D \* C \* A \* B'  
+ F' \* E' \* D' \* C' \* A' \* B  
+ F \* E' \* D' \* C' \* A' \* B B  
+ F' \* E \* D \* C' \* A' \* B B  
+ F \* E' \* D' \* C \* A' \* B B  
+ F' \* E' \* D \* C \* A' \* B B  
+ F \* E' \* D' \* C' \* A \* B  
+ F' \* E' \* D \* C' \* A \* B B  
+ F \* E' \* D \* C' \* A \* B  
+ F' \* E' \* D' \* C \* A \* B  
+ F \* E \* D' \* C \* A \* B  
+ F' \* E' \* D \* C \* A \* B  
+ F' \* E \* D \* C \* A \* B;

ENDS

292003-5

Cascading XOR gates--  
combinatorial feedback used  
LEF Version 2.0 Baseline 4.3i  
22 Jul 1988

PART: 85C220

INPUTS: A, B, C, D, E, F  
OUTPUTS: O, NODE

**NETWORK:**

```
A = INP(A)
B = INP(B)
C = INP(C)
D = INP(D)
E = INP(E)
F = INP(F)
O = CONF(OC, VCC)
NODE, NODE = COIF(NODEC, VCC)
```

EQUATIONS:

$$\begin{aligned} \text{NODEC} &= D * C' * A' * B' \\ &+ D' * C * A' * B \\ &+ D' * C' * A * B' \\ &+ D * C * A * B' \\ &+ D' * C' * A' * B \\ &+ D * C * A' * B \\ &+ D' * C' * A * B \\ &+ D' * C * A * B; \end{aligned}$$
$$\begin{aligned} \text{Oc} &= F * E' * \text{NODE}' \\ &+ F' * E * \text{NODE}' \\ &+ F' * E' * \text{NODE} \\ &+ F * E * \text{NODE}; \end{aligned}$$

ENDS

292003-6

3

### Figure 3. Minimized Logic Equations for Figure 1

**Figure 4. Minimized Logic Equations for Figure 3**

February 1987

# 16-Bit Binary Counter Implementation Using the 5C060 EPLD

KARL-HEINZ WEIGL  
INTEL CORPORATION  
MUNICH, GERMANY

# 16-BIT BINARY COUNTER IMPLEMENTATION USING THE 5C060 EPD

Additional product term for every successive significant bit, whereas toggle flip-flop implementation requires only one product term per significant bit. Thus, the toggle flip-flop counter design is more concise in product term consumption than the D register design. Since product term minimization is the key element to maximizing PLA utilization, the T-FF counter design is more efficient. The truth table for the toggle flip-flop is shown in Fig. 2.

T	Q(N)	Q(N + 1)
0	0	0
0	1	1
1	0	1
1	1	0

Figure 2

## SOLUTION

The 16-bit binary counter function was implemented in the 5C060 EPD using the Lant Programmable Logic Development System (PLDS). The equations for the 16-bit binary counter with the RESET, UP/DOWN, and RUN/STOP functions are shown in the EQUATIONS section of the LEP (Fig. 4). The pinout of the 5C060 with the implemented counter is shown in the RPT file (Utilization Report) Fig. 5. This RPT file also shows under the OUTPUTS section that in each macrocell only one out of 8 product terms is used. In contrast the same 16-bit counter designed using D-type flip-flops would have required more than 16 product terms for the last significant bit.

CONTENTS	PAGE
INTRODUCTION	3-140
DESIGN OBJECTIVE	3-140
SOLUTION	3-140

The 5C060 EPD is a 16-bit binary counter with 16 outputs. The 5C060 EPD is a 16-bit binary counter with 16 outputs. The 5C060 EPD is a 16-bit binary counter with 16 outputs.

Use of traditional 20-pin and 24-pin PLAs, however, does not allow for the construction of large counters having greater than 10 significant bits. This is because these traditional PLAs have register and product term restrictions (even the largest bipolar PLAs have only 8 to 10 registers and less than 8 product terms per register). In contrast, the 5C060 24-pin erasable programmable logic device (EPD) contains 16 registers that are programmable as D, T, R2, or JK types. These 16 programmable registers enable the construction of 16-bit counters with up to 16 significant bits.

The application brief details the implementation of a 16-bit binary counter in the 5C060 EPD. The design also demonstrates efficient construction utilizing toggle flip-flops (T-FF) that allows for minimum product term utilization.

## DESIGN OBJECTIVE

The objective of the design is to implement a counter with the following features: (i) 16-bit binary count, (ii) toggle flip-flop, (iii) asynchronous clear, (iv) RUN/STOP function, and (v) UP/DOWN function. The function table is shown in Figure 1.

Function	RESET	UP/DOWN	RUN/STOP
Initial Counting	0	X	X
Count Down	1	0	0
Count Up	1	1	0
Reset All Outputs to "LOW"	X	X	1

Figure 1



## INTRODUCTION

System designers often use programmable logic devices to implement counters. Use of PLA devices lets the user build customized counters to suit individual applications. In most cases such counters are not available, 'off-the-shelf' SSI/MSI devices. In other applications, the PLA implementation allows the designer to squeeze the counter function along with other 'glue' tasks into a single PLA, with the attendant higher integration benefits.

Use of traditional 20-pin and 24-pin PLAs, however, does not allow for the construction of large counters having greater than 10 significant bits. This is because these traditional PLAs have register and product term restrictions (even the larger bipolar PLAs have only 8 to 10 registers and less than 8 product terms per register). In contrast, the 5C060 24-pin erasable programmable logic device (EPLD) contains 16 registers that are programmable as 'D', 'T', 'RS' or 'JK' types. These 16 programmable registers enable the construction of Up/Down counters with up to 16 significant bits.

This application brief details the implementation of a 16-bit binary counter in the 5C060 EPLD. The design also demonstrates efficient counter construction utilizing toggle flip-flops (T-FF) that allows for minimum product term utilization.

## DESIGN OBJECTIVE

The objective of the design is to implement a counter with the following features: (i) 16-bit binary count, (ii) toggle flip-flops, (iii) asynchronous clear, (iv) RUN/STOP function and (v) UP/DOWN function. The function table is shown in Figure 1.

RESET	UP/DOWN	RUN/STOP	Function
X	X	0	Inhibit Counting
0	0	1	Count Down
0	1	1	Count Up
1	X	X	Reset All Outputs to 'LOW'

Figure 1

## TOGGLE FLIP-FLOPS

Counters can be most effectively implemented in PLA architectures using toggle flip-flops. This is because counters constructed with 'D' type flip-flops require an additional product term for every successive significant bit, whereas toggle flip-flop implementation requires only **one** product term per significant bit. Thus, the toggle flip-flop counter design is more miserly in product term consumption than the 'D' register design. Since product term minimization is the key element to maximizing PLA utilization, the T-FF counter design is more efficient. The truth table for the toggle flip-flop is shown in Fig. 2.

T	Q(N)	Q (N + 1)
0	0	0
0	1	1
1	0	1
1	1	0

Figure 2

## SOLUTION

The 16-bit binary counter function was implemented in the 5C060 EPLD using the Intel Programmable Logic Development System (iPLDS). The equations for the 16-bit binary counter with the RESET, UP/DOWN and RUN/STOP functions are shown in the 'EQUATIONS' section of the LEF (Fig. 4). The pinout of the 5C060 with the implemented counter is shown in the RPT file (Utilization Report) Fig. 5. This RPT file also shows, under the 'OUTPUTS' section, that in each macrocell only one out of 8 product terms is used. In contrast the same 16-bit counter designed using 'D' type flip-flops would have required more than 16 product terms for the last significant bit.



```

Q1U = AND (UD,Q0F,Q0U)
Q2U = AND (UD,Q1F,Q1U)
Q3U = AND (UD,Q2F,Q2U)
Q4U = AND (UD,Q3F,Q3U)
Q5U = AND (UD,Q4F,Q4U)
Q6U = AND (UD,Q5F,Q5U)
Q7U = AND (UD,Q6F,Q6U)
Q8U = AND (UD,Q7F,Q7U)
Q9U = AND (UD,Q8F,Q8U)
QAU = AND (UD,Q9F,Q9U)
QBU = AND (UD,QAF,QAU)
QCU = AND (UD,QBF,QBU)
QDU = AND (UD,QCF,QCU)
QEU = AND (UD,QDF,QDU)
QFU = AND (UD,QEF,QEU)
NQ0F = NOT (Q0F)
NQ1F = NOT (Q1F)
NQ2F = NOT (Q2F)
NQ3F = NOT (Q3F)
NQ4F = NOT (Q4F)
NQ5F = NOT (Q5F)
NQ6F = NOT (Q6F)
NQ7F = NOT (Q7F)
NQ8F = NOT (Q8F)
NQ9F = NOT (Q9F)
NQAF = NOT (QAF)
NQBF = NOT (QBF)
NQCF = NOT (QCF)
NQDF = NOT (QDF)
NQEF = NOT (QEF)
Q0D = AND (NUD,RS)
Q1D = AND (NUD,NQ0F,Q0D)
Q2D = AND (NUD,NQ1F,Q1D)
Q3D = AND (NUD,NQ2F,Q2D)
Q4D = AND (NUD,NQ3F,Q3D)
Q5D = AND (NUD,NQ4F,Q4D)
Q6D = AND (NUD,NQ5F,Q5D)
Q7D = AND (NUD,NQ6F,Q6D)
Q8D = AND (NUD,NQ7F,Q7D)
Q9D = AND (NUD,NQ8F,Q8D)
QAD = AND (NUD,NQ9F,Q9D)
QBD = AND (NUD,NQAF,QAD)
QCD = AND (NUD,NQBF,QBD)
QDD = AND (NUD,NQCF,QCD)
QED = AND (NUD,NQDF,QDD)
QFD = AND (NUD,NQEF,QED)
ENDS

```

292015-2

Figure 3. Example .ADF (Continued)

```

INTEL CORPORATION
JAN. 15, 1987
1
1.0
5C060
BINARY 16-BIT UP/DOWN COUNTER WITH RUN/STOP AND ASYNCH. RESET USING T-FF

LB Version 4.01, Baseline 27.1 4/9/86
LEF Version 4.01 Baseline 22.2 2/4/86
OPTIONS: TURBO=ON
PART:
5C060

INPUTS:
RS, CLOCK, RESET, UD

OUTPUTS:
Q0, Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8, Q9, QA, QB, QC, QD, QE, QF

NETWORK:
CLK = INP(CLOCK)
RS = INP(RS)
CLR = INP(RESET)
UD = INP(UD)
Q0, Q0F = TOTF(Q0T, CLK, CLR, GND, VCC)
Q1, Q1F = TOTF(Q1T, CLK, CLR, GND, VCC)
Q2, Q2F = TOTF(Q2T, CLK, CLR, GND, VCC)
Q3, Q3F = TOTF(Q3T, CLK, CLR, GND, VCC)
Q4, Q4F = TOTF(Q4T, CLK, CLR, GND, VCC)
Q5, Q5F = TOTF(Q5T, CLK, CLR, GND, VCC)
Q6, Q6F = TOTF(Q6T, CLK, CLR, GND, VCC)
Q7, Q7F = TOTF(Q7T, CLK, CLR, GND, VCC)
Q8, Q8F = TOTF(Q8T, CLK, CLR, GND, VCC)
Q9, Q9F = TOTF(Q9T, CLK, CLR, GND, VCC)
QA, QAF = TOTF(QAT, CLK, CLR, GND, VCC)
QB, QBF = TOTF(QBT, CLK, CLR, GND, VCC)
QC, QCF = TOTF(QCT, CLK, CLR, GND, VCC)
QD, QDF = TOTF(QDT, CLK, CLR, GND, VCC)
QE, QEF = TOTF(QET, CLK, CLR, GND, VCC)
QF = TONF(QFT, CLK, CLR, GND, VCC)

EQUATIONS:
QFT = UD' * QEF' * QDF' * QCF' * QBF' * QAF' * Q9F' * Q8F' * Q7F' * Q6F' * Q5F' * Q4F' * Q3F' * Q2F' * Q1F' * Q0F' * RS
      + UD * QEF * QDF * QCF * QBF * QAF * Q9F * Q8F * Q7F * Q6F * Q5F * Q4F * Q3F * Q2F * Q1F * Q0F * RS;

QET = UD' * QDF' * QCF' * QBF' * QAF' * Q9F' * Q8F' * Q7F' * Q6F' * Q5F' * Q4F' * Q3F' * Q2F' * Q1F' * Q0F' * RS
      + UD * QDF * QCF * QBF * QAF * Q9F * Q8F * Q7F * Q6F * Q5F * Q4F * Q3F * Q2F * Q1F * Q0F * RS;

QDT = UD' * QCF' * QBF' * QAF' * Q9F' * Q8F' * Q7F' * Q6F' * Q5F' * Q4F' * Q3F' * Q2F' * Q1F' * Q0F' * RS
      + UD * QCF * QBF * QAF * Q9F * Q8F * Q7F * Q6F * Q5F * Q4F * Q3F * Q2F * Q1F * Q0F * RS;

```

292015-3

Figure 4. Example .LEF



```

QCT = UD' * QBF' * QAF' * Q9F' * Q8F' * Q7F' * Q6F' * Q5F' * Q4F' * Q3F' *
      Q2F' * Q1F' * Q0F' * RS
+ UD * QBF * QAF * Q9F * Q8F * Q7F * Q6F * Q5F * Q4F * Q3F * Q2F *
  Q1F * Q0F * RS;

QBT = UD' * QAF' * Q9F' * Q8F' * Q7F' * Q6F' * Q5F' * Q4F' * Q3F' * Q2F' *
      Q1F' * Q0F' * RS
+ UD * QAF * Q9F * Q8F * Q7F * Q6F * Q5F * Q4F * Q3F * Q2F * Q1F *
  Q0F * RS;

QAT = UD' * Q9F' * Q8F' * Q7F' * Q6F' * Q5F' * Q4F' * Q3F' * Q2F' * Q1F' *
      Q0F' * RS
+ UD * Q9F * Q8F * Q7F * Q6F * Q5F * Q4F * Q3F * Q2F * Q1F * Q0F *
  RS;

Q9T = UD' * Q8F' * Q7F' * Q6F' * Q5F' * Q4F' * Q3F' * Q2F' * Q1F' * Q0F' *
      RS
+ UD * Q8F * Q7F * Q6F * Q5F * Q4F * Q3F * Q2F * Q1F * Q0F * RS;

Q8T = UD' * Q7F' * Q6F' * Q5F' * Q4F' * Q3F' * Q2F' * Q1F' * Q0F' * RS
+ UD * Q7F * Q6F * Q5F * Q4F * Q3F * Q2F * Q1F * Q0F * RS;

Q7T = UD' * Q6F' * Q5F' * Q4F' * Q3F' * Q2F' * Q1F' * Q0F' * RS
+ UD * Q6F * Q5F * Q4F * Q3F * Q2F * Q1F * Q0F * RS;

Q6T = UD' * Q5F' * Q4F' * Q3F' * Q2F' * Q1F' * Q0F' * RS
+ UD * Q5F * Q4F * Q3F * Q2F * Q1F * Q0F * RS;

Q5T = UD' * Q4F' * Q3F' * Q2F' * Q1F' * Q0F' * RS
+ UD * Q4F * Q3F * Q2F * Q1F * Q0F * RS;

Q4T = UD' * Q3F' * Q2F' * Q1F' * Q0F' * RS
+ UD * Q3F * Q2F * Q1F * Q0F * RS;

Q3T = UD' * Q2F' * Q1F' * Q0F' * RS
+ UD * Q2F * Q1F * Q0F * RS;

Q2T = UD' * Q1F' * Q0F' * RS
+ UD * Q1F * Q0F * RS;

Q1T = UD' * Q0F' * RS
+ UD * Q0F * RS;

Q0T = RS;
ENDS

```

292015-4

Figure 4. Example .LEF (Continued)

int 1

FIT Version 4.01 Baseline 27.1 4/9/86

\*\*\*\*\* Design implemented successfully

\*\*\*\* NOTE: Connect signal CLOCK to pin 1 AND pin 13.

INTEL CORPORATION

JAN. 15, 1987

1

1.0

5C060

BINARY 16-BIT UP/DOWN COUNTER WITH RUN/STOP AND ASYNCH. RESET USING T-FF

LB Version 4.01, Baseline 27.1 4/9/86

OPTIONS: TURBO=ON

5C060

```
CLOCK -- 1 24:- Vcc
GND -- 2 23:- RS
Q7 -- 3 22:- QF
Q6 -- 4 21:- QE
Q5 -- 5 20:- QD
Q4 -- 6 19:- QC
Q3 -- 7 18:- QB
Q2 -- 8 17:- QA
Q1 -- 9 16:- Q9
Q0 -- 10 15:- Q8
UD -- 11 14:- RESET
GND -- 12 13:- CLOCK
```

\*\*INPUTS\*\*

Name	Pin	Resource	MCell #	PTerms	MCells	Feeds:	OE	Clear	Clock
CLOCK	1	INP	-	-	-	-	-	-	CLK1 CLK2
UD	11	INP	-	-	1	-	-	-	-
					2				
					3				
					4				
					5				
					6				
					7				
					8				
					9				
					10				
					11				
					12				
					13				
					14				
					15				
GND	12	GND	-	-	-	-	-	-	-
CLOCK	13	INP	-	-	-	-	-	-	CLK1 CLK2
RESET	14	INP	-	-	-	-	-	1	-
								2	
								3	
								4	
								5	
								6	
								7	
								8	
								9	
								10	
								11	
								12	
								13	
								14	
								15	
								16	

292015-5

Figure 5. Example .RPT File

**Figure 5. Example .RPT File (Continued)**

Q3	7	TOTF	13	2/ 8	1	-	-	-	TOTF	81	99
					2						
					3						
					4						
					5						
					6						
					7						
					8						
					9						
					10						
					11						
					12						
Q2	8	TOTF	14	2/ 8	1	-	-	-	TOTF	82	99
					2						
					3						
					4						
					5						
					6						
					7						
					8						
					9						
					10						
					11						
					12						
					13						
Q1	9	TOTF	15	2/ 8	1	-	-	-	TOTF	83	99
					2						
					3						
					4						
					5						
					6						
					7						
					8						
					9						
					10						
					11						
					12						
					13						
Q0	10	TOTF	16	1/ 8	1	-	-	-	TOTF	84	99
					2						
					3						
					4						
					5						
					6						
					7						
					8						
					9						
					10						
					11						
					12						
					13						
					14						
Q8	15	TOTF	8	2/ 8	1	-	-	-	TOTF	85	99
					2						
					3						
					4						
					5						
					6						
					7						
Q9	16	TOTF	7	2/ 8	1	-	-	-	TOTF	86	99
					2						
					3						
					4						
					5						
					6						
QA	17	TOTF	6	2/ 8	1	-	-	-	TOTF	87	99
					2						
					3						
					4						
					5						

292015-7

Figure 5. Example .RPT File (Continued)



**Figure 5. Example .RPT File (Continued)**

3-151	INTRODUCTION
3-151	SC033 MAILBOX
3-152	SC033 MAILBOX CONTROLLER
3-153	Block Diagram
3-154	SC033 "Back to Back Register"
3-155	SC033 "Mailbox Controller"
3-156	SC033 Register ADF
3-157	SC033 Address ADF

## MEMORY FOR TWO 80C31 MICROCONTROLLERS USING EPLDs

October 1989

3

# Designing a Mailbox Memory for Two 80C31 Microcontrollers Using EPLDs

**K. WEIGL & J. STAHL**  
INTEL CORPORATION  
MUNICH, GERMANY

Order Number: 292016-004

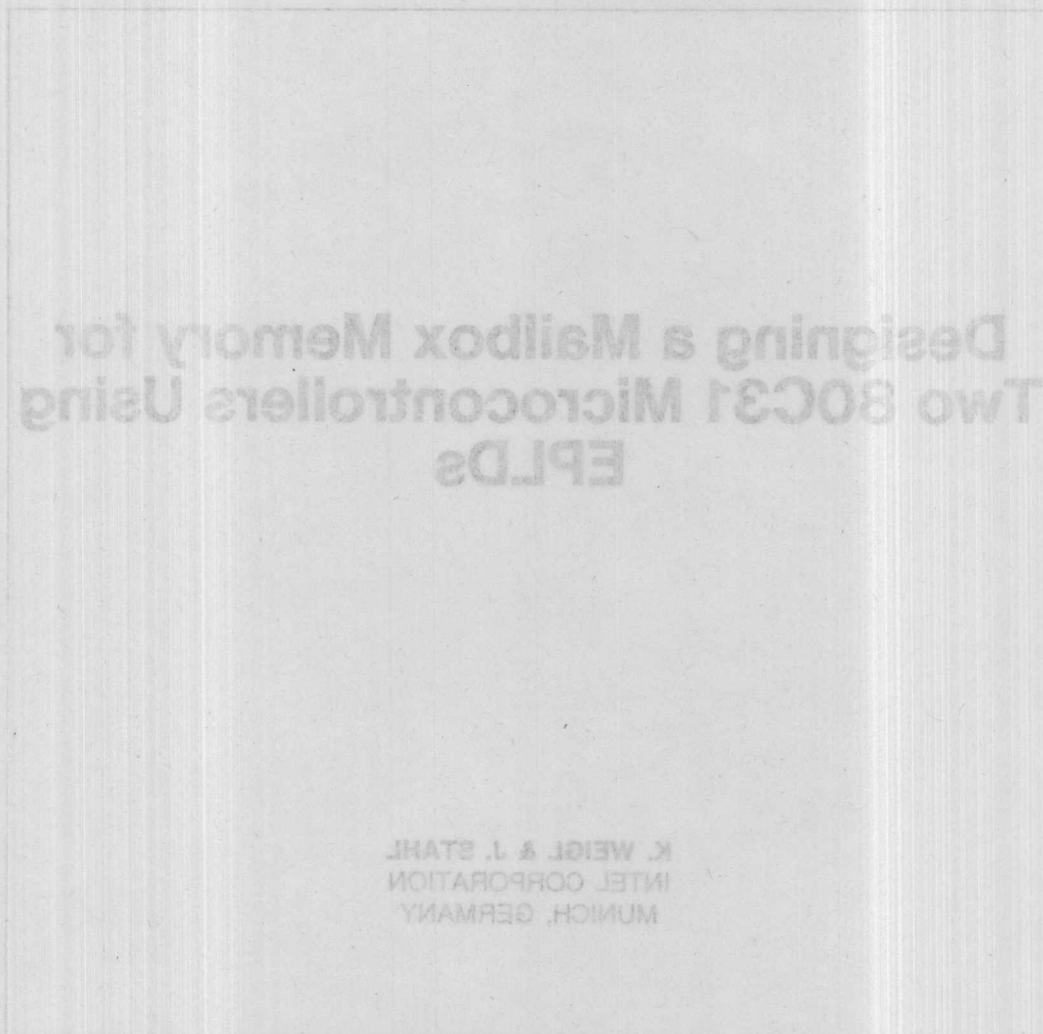
**DESIGNING A MAILBOX  
MEMORY FOR TWO 80C31  
MICROCONTROLLERS  
USING EPLDs**

**CONTENTS**

PAGE

INTRODUCTION .....	3-151
5C060 MAILBOX .....	3-151
5C032 MAILBOX CONTROLLER .....	3-152
Block Diagram .....	3-153
5C060 "Back to Back Register" .....	3-154
5C032 "Mailbox Controller" .....	3-155
5C060 Register ADF .....	3-156
5C032 Arbiter ADF .....	3-157

October 1989



K. WEIGL & J. STAHL  
INTEL CORPORATION  
MUNICH, GERMANY

Order Number: 332019-004

## INTRODUCTION

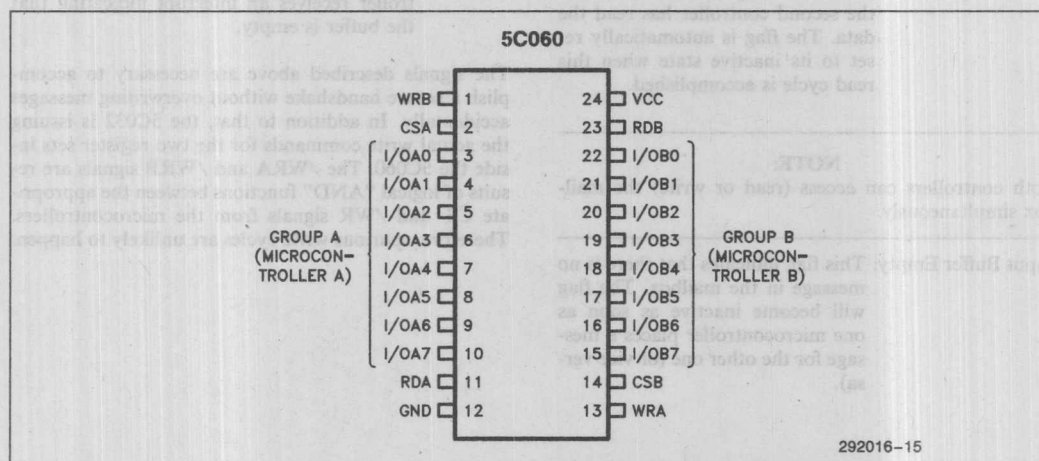
Very often, complex systems involve two or more microcontrollers to fulfill the requirements defined by a given objective. Since the nature of microcontrollers does not allow for easy dual-port memory design (no "READY" input; no "HOLD/HLDA" interface; port-oriented I/O etc.), design engineers are faced with the problem of interchanging information (data and status) between those microcontrollers. This application brief describes the design of a mailbox for exchanging information between two 80C31s, using a 5C060 EPLD as a "back-to-back" register, and a 5C032 EPLD as an arbitration vehicle to control the actions of the CPUs.

## THE 5C060 MAILBOX

In this application, the 16 macrocells of the 5C060 are grouped into two sets of 8 so called "ROIF" (register output with input feedback) primitives to implement the two 8 bit bus interfaces needed. The grouping is done according to the following picture.

The 5C060 allows for independent clocking of 8 macrocells on each side of the chip, the two clock inputs are used to clock data from the microcontroller bus into the chip. To read the data written into the mailbox by one of the controllers, the RDA- (controller A is reading) or RDB- (controller B is reading) line must be pulled low by activating the read command (/RD). In order to avoid spurious read-cycles, the /RD commands from both microcontrollers are logically "ORed" together with an active high CS-signal (Chip Select) inside the 5C060. The CS-signal for both ports is derived from address line A15. Therefore, whenever A15 becomes a logic "1" (true), the mailbox is activated and ready to take or submit data.

Address range for the mailbox: F000 Hex to FFFF Hex  
(Upper 12 kbyte)





## THE 5C032 "MAILBOX CONTROLLER"

To keep the two microcontrollers informed about the status of their mailbox, the 5C032 is programmed to supply the following signals to both controllers:

**/OBFA: "OUTPUT BUFFER FULL" FOR MC A**

**/OBFB: "OUTPUT BUFFER FULL" FOR MC B**

**/IBEA: "INPUT BUFFER EMPTY" FOR MC A**

**/IBEB: "INPUT BUFFER EMPTY" FOR MC B**

**/INTA: INTERRUPT TO MC A**

**/INTB: INTERRUPT TO MC B**

The next section will discuss the meanings of these signals in more detail.

**Output Buffer Full:** This flag is set whenever the controller writes into its own output buffer. The flag remains valid, until the second controller has read the data. The flag is automatically reset to its inactive state when this read cycle is accomplished.

### NOTE:

Both controllers can access (read or write) the mailbox simultaneously.

**Input Buffer Empty:** This flag indicates that there is no message in the mailbox. The flag will become inactive as soon as one microcontroller places a message for the other one (or vice versa).

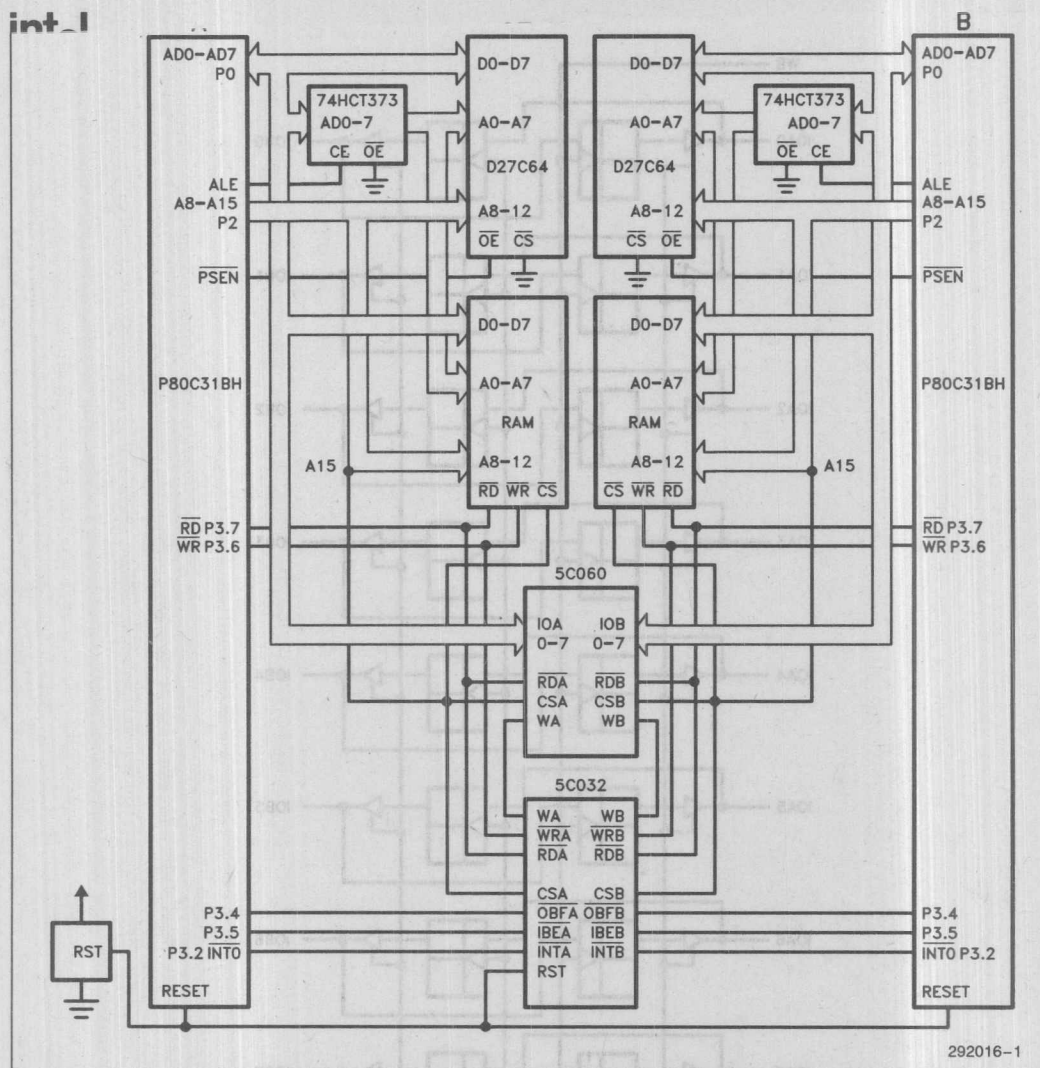
Example: /IBEA remains "LOW" until microcontroller B places a message for controller A into the mailbox for A. /IBEA will go "HIGH" as soon as controller B has accomplished its write cycle, and will not go "LOW" again until microcontroller A has read the message.

**Interrupt:** The 5C032 is programmed to supply interrupts to both microcontrollers involved, on one of the following events.

1. The /OBF flag of the opposite microcontroller becomes active; e.g. if controller A is placing a message for controller B, controller B receives an interrupt the same time as /OBFA becomes valid or vice versa.
2. The /IBE flag of the opposite microcontroller goes active, indicating that this controller has received the message; e.g. if controller B reads the message stored by controller A, its /IBEB flag goes active and controller receives an interrupt indicating that the buffer is empty.

The signals described above are necessary to accomplish a secure handshake without overwriting messages accidentally. In addition to that, the 5C032 is issuing the actual write commands for the two register sets inside the 5C060. The /WRA and /WRB signals are results of logical "AND" functions between the appropriate CS- and /WR signals from the microcontrollers. Therefore, spurious write cycles are unlikely to happen.

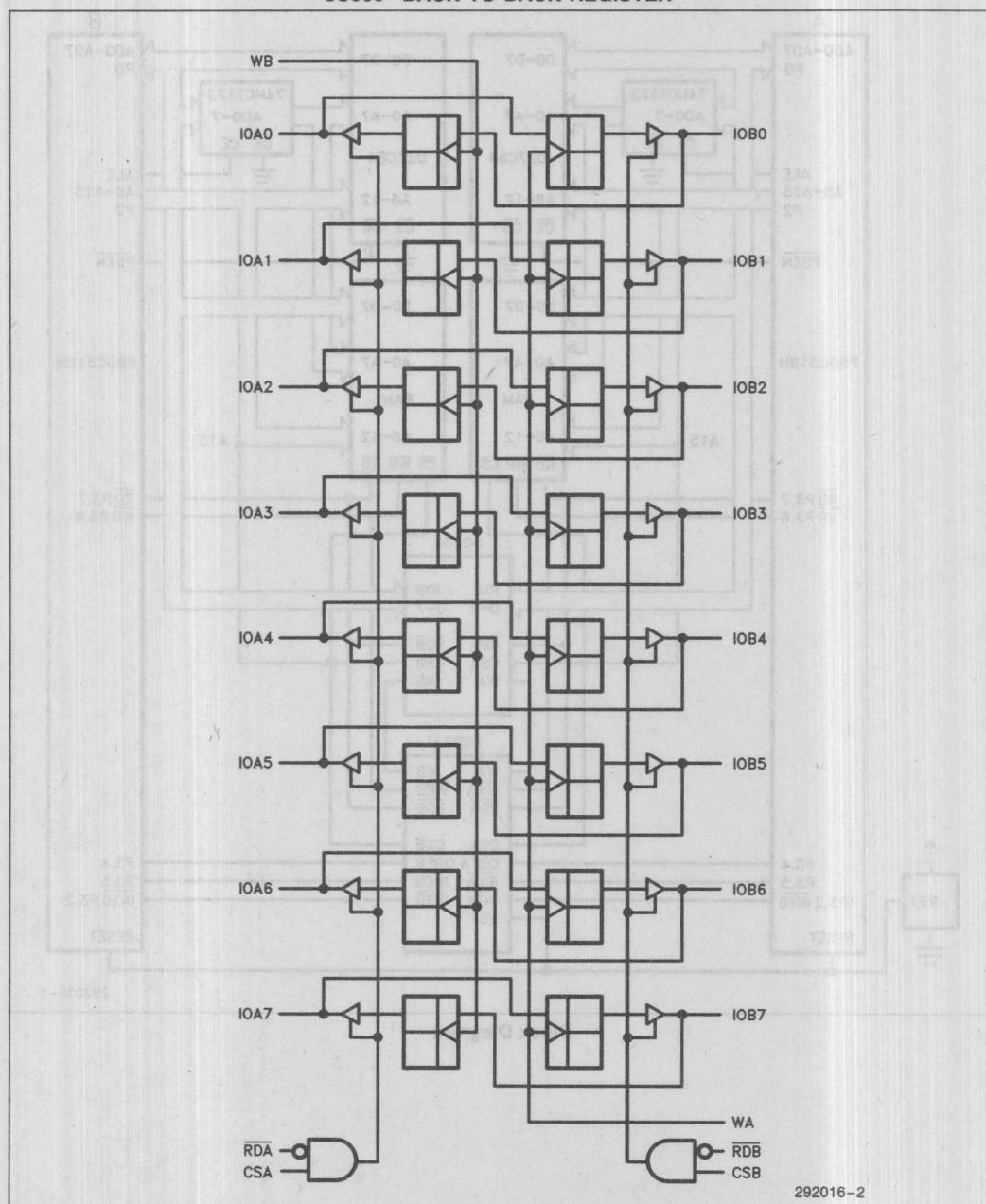
int-1



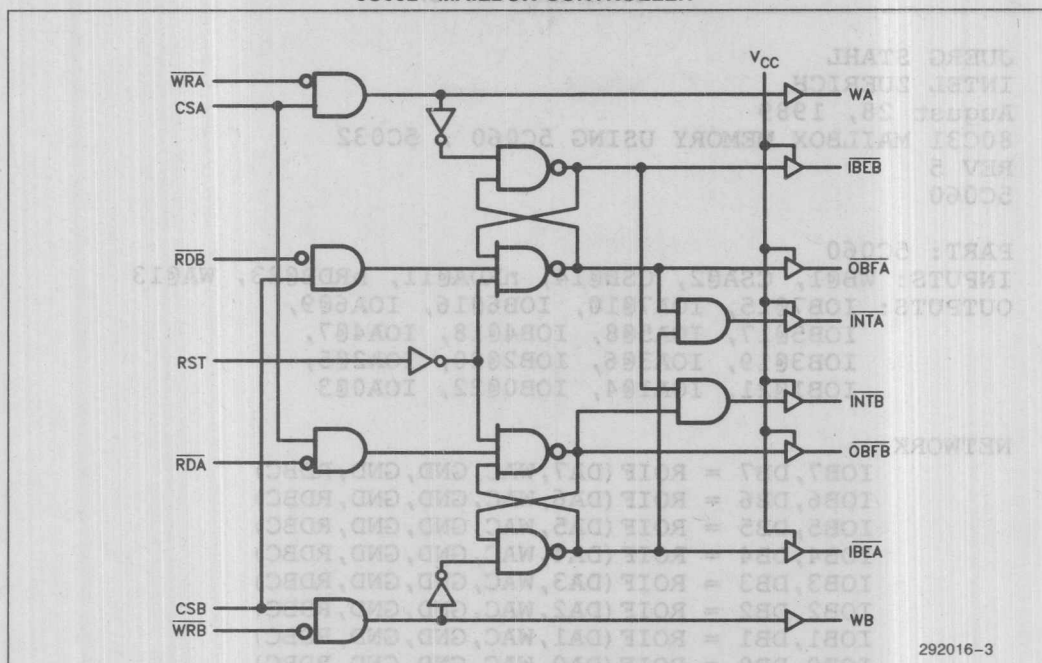
Block Diagram

3

## 5C060 "BACK TO BACK REGISTER"



5C032 "MAILBOX CONTROLLER"



3



## 5C060 REGISTER ADF

JUERG STAHL  
 INTEL ZUERICH  
 August 28, 1989  
 80C31 MAILBOX MEMORY USING 5C060 / 5C032  
 REV 5  
 5C060

PART: 5C060

INPUTS: WB@1, CSA@2, CSB@14, nRDA@11, nRDB@23, WA@13

OUTPUTS: IOB7@15, IOA7@10, IOB6@16, IOA6@9,  
 IOB5@17, IOA5@8, IOB4@18, IOA4@7,  
 IOB3@19, IOA3@6, IOB2@20, IOA2@5,  
 IOB1@21, IOA1@4, IOB0@22, IOA0@3

NETWORK:

IOB7,DB7 = ROIF(DA7,WAC,GND,GND,RDBC)  
 IOB6,DB6 = ROIF(DA6,WAC,GND,GND,RDBC)  
 IOB5,DB5 = ROIF(DA5,WAC,GND,GND,RDBC)  
 IOB4,DB4 = ROIF(DA4,WAC,GND,GND,RDBC)  
 IOB3,DB3 = ROIF(DA3,WAC,GND,GND,RDBC)  
 IOB2,DB2 = ROIF(DA2,WAC,GND,GND,RDBC)  
 IOB1,DB1 = ROIF(DA1,WAC,GND,GND,RDBC)  
 IOB0,DB0 = ROIF(DA0,WAC,GND,GND,RDBC)  
 IOA7,DA7 = ROIF(DB7,WBC,GND,GND,RDAC)  
 IOA6,DA6 = ROIF(DB6,WBC,GND,GND,RDAC)  
 IOA5,DA5 = ROIF(DB5,WBC,GND,GND,RDAC)  
 IOA4,DA4 = ROIF(DB4,WBC,GND,GND,RDAC)  
 IOA3,DA3 = ROIF(DB3,WBC,GND,GND,RDAC)  
 IOA2,DA2 = ROIF(DB2,WBC,GND,GND,RDAC)  
 IOA1,DA1 = ROIF(DB1,WBC,GND,GND,RDAC)  
 IOA0,DA0 = ROIF(DB0,WBC,GND,GND,RDAC)  
 WAC = INP(WA)  
 WBC = INP(WB)  
 CSB = INP(CSB)  
 CSA = INP(CSA)  
 nRDB = INP(nRDB)  
 nRDA = INP(nRDA)

EQUATIONS:

RDBC = CSB \* !nRDB;

RDAC = CSA \* !nRDA;

END\$

JUERG STAHL  
 INTEL ZUERICH  
 August 28, 1989  
 80C31 MAILBOX MEMORY USING 5C060 / 5C032  
 REV 5  
 5C032

PART: 5C032

INPUTS: RST, nWRA, nRDB, CSA, nRDA, nWRB, CSB

OUTPUTS: WA, nOBFA, nIBEB, nINTA, nINTB, nOBFB, nIBEA, WB

NETWORK:

nWRA = INP (nWRA)  
 nRDA = INP (nRDA)  
 CSA = INP (CSA)  
 nWRB = INP (nWRB)  
 nRDB = INP (nRDB)  
 CSB = INP (CSB)  
 RST = INP (RST)  
 WA = CONF (WAd, VCC)  
 WB = CONF (WBd, VCC)  
 nOBFA, nOBFA = COIF (nOBFA, VCC)  
 nOBFB, nOBFB = COIF (nOBFB, VCC)  
 nIBEA, nIBEA = COIF (nIBEA, VCC)  
 nIBEB, nIBEB = COIF (nIBEB, VCC)  
 nINTA = CONF (nINTA, VCC)  
 nINTB = CONF (nINTB, VCC)

EQUATIONS:

nINTBd = nOBFB \* nIBEB;  
 nINTAd = nOBFA \* nIBEA;  
 nOBFBd = !(!(nRDA \* CSA) \* nIBEA \* !RST);  
 nOBFA = !(!(nRDB \* CSB) \* nIBEB \* !RST);  
 nIBEBd = !(!(CSA \* !nWRA) \* nOBFA);  
 nIBEA = !(!(CSB \* !nWRB) \* nOBFB);  
 WAd = CSA \* !nWRA;  
 WBd = CSB \* !nWRB;

END\$

292016-9

October 1988

# Atypical Latch/Register Construction in EPLDs

THOM BOWNS

PROGRAMMABLE LOGIC APPLICATIONS  
INTEL CORPORATION

Order Number: 292031-003

# ATYPICAL LATCH/ REGISTER CONSTRUCTION IN EPLDs

we must convert the equation to a logic form with feedback.

$$Q_2 \text{ or } Q_1 = Q_2 \text{ or } Q_1 \text{ or } Q_2$$

$$Q = S + R \cdot Q$$

where  $Q$  is the feedback from  $Q$  output.

This circuit can be implemented in an EPLD macro-cell. When combinational feedback is not supported, I/O feedback will suffice. The schematic of this implementation is shown in Figure 1b.

With the RS latch, the inputs are normally low. A high on  $S$  sets  $Q$  to 1, and a high on  $R$  resets  $Q$  to 0. If both  $S$  and  $R$  are high simultaneously, the output is undefined. In this implementation, the output is forced to remain at a high level since  $S$  takes priority over  $R$ .

## CONTENTS

PAGE

### REGISTERS/LATCHES AVAILABLE IN EPLDs

The RS Latch	3-160
The D Latch	3-161
The D Flip Flop	3-162
The D Flip Flop (continued)	3-163
The RS Flip Flop	3-164
The JK Flip Flop	3-164
The T Flip Flop	3-165

The RS latch is the simplest latch configuration. The equations for  $S$  and  $R$  are as follows:  $Q = S + R \cdot Q$ .  $Q$  is the output of the other  $Q$  input.  $Q$  is the output of the other  $Q$  input.

3

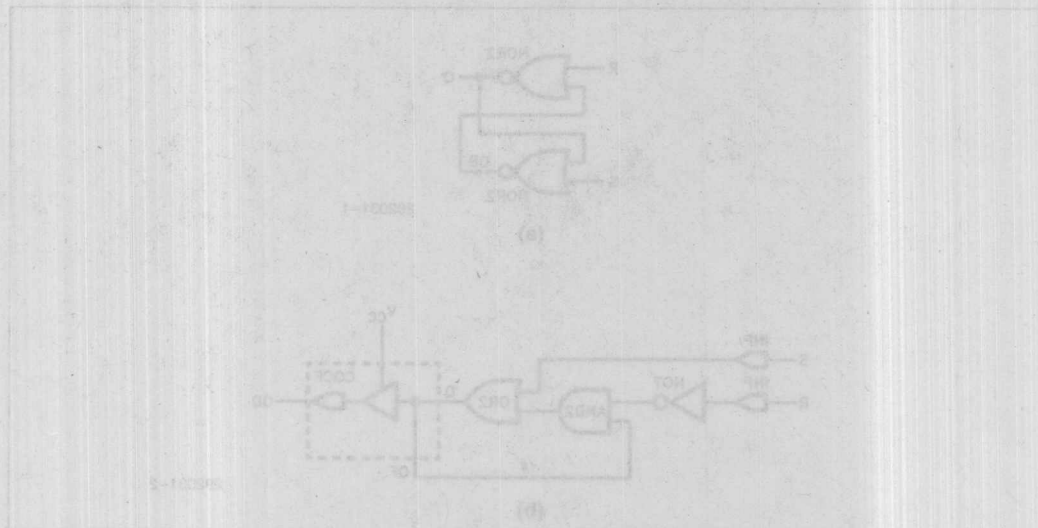


Figure 1. RS Latch implementation in a) Direct Gate and b) EPLD logic



## ATYPICAL LATCH/REGISTER CONSTRUCTION IN EPLDs

Though Intel's EPLDs include many of the typical latch and register types, some logic designs require register or latch configurations not directly supported in the current EPLDs. In many cases these register and latch configurations can be generated using the logic array and combinational feedback. A "latch" is defined as a level-triggered, flow-through type such as the 74373, and a "register" is defined as an edge-triggered flip-flop such as the 7474.

This application brief will detail the construction of a D-type latch, an RS latch and a D flip-flop using combinational logic and feedback. Also discussed is the construction of an RS flip-flop, a JK flip-flop and a T flip-flop using registered logic and feedback.

The RS latch is the simplest latch configuration. The equations for it are as follows:  $QB = \overline{(Q + S)}$ ,  $Q = \overline{(QB + R)}$  where Q is the output of one NOR gate, and QB is the output of the other (Note: as a convention

in this Ap brief, the "!" operator is used to signify inversion). The schematic of the RS latch is shown in Figure 1a.

Since cross coupled logic is not supported in EPLDs, we must convert the equation to a single term with feedback.

$$QD, QF = COCF(Q, VCC)$$

$$Q = S + \overline{!R} * QF;$$

where QF is the feedback from Q output.

This circuit can be implemented in an EPLD macrocell. Where combinational feedback is not supported, I/O feedback will suffice. The schematic of this implementation is shown in Figure 1b.

With the RS latch, the inputs are normally low. A logical one on S sets Q to 1, and a one on R resets Q to a 0. Logical ones on both inputs simultaneously cause the output to remain at a high level since S takes precedence over R in this implementation.

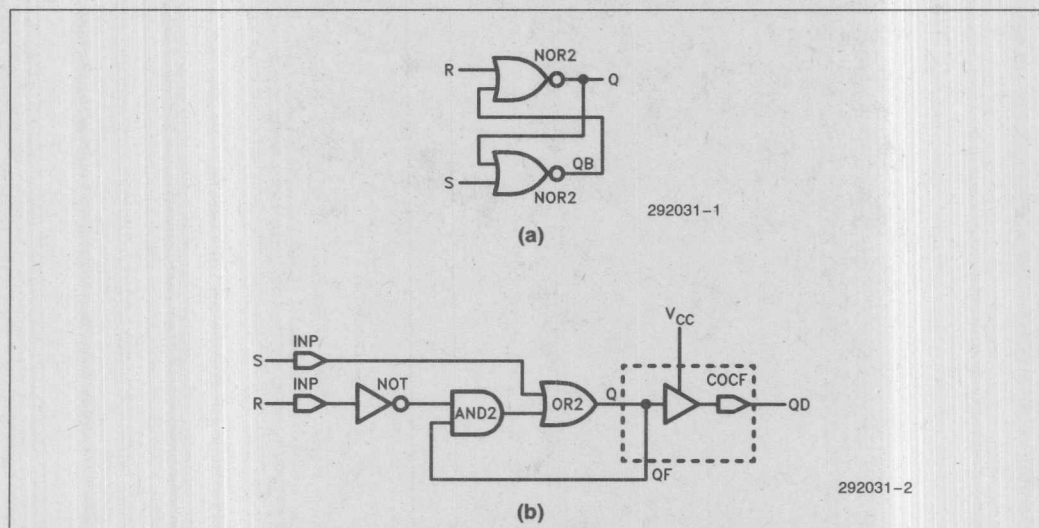


Figure 1. RS Latch Implementation In a) Discrete Gates and b) EPLD Logic

Another latch is the 143/5 type, or D latch. This latch works by either enabling input data to appear at the output, or by holding the output to the last input data state. Its equation is this:  $QB = !(!D * E) * Q$ ,  $Q = !(!D * E) * QB$ . Again, Q is the output of one NAND gate, and QB is the output of the other. Figure 2a shows this version of the design.

Again, we must convert to an EPLD-type equation and schematic:

$$QD, QF = COCF(Q, VCC)$$

$$Q = D * E + !E * QF;$$

QF is the feedback from the COCF. In this circuit, when E is high, data flows through transparently. When E is brought low, data is latched. When using input feedback, care must be taken when tri-stating the output as data will no longer be latched. The EPLD implementation is given in Figure 2b.

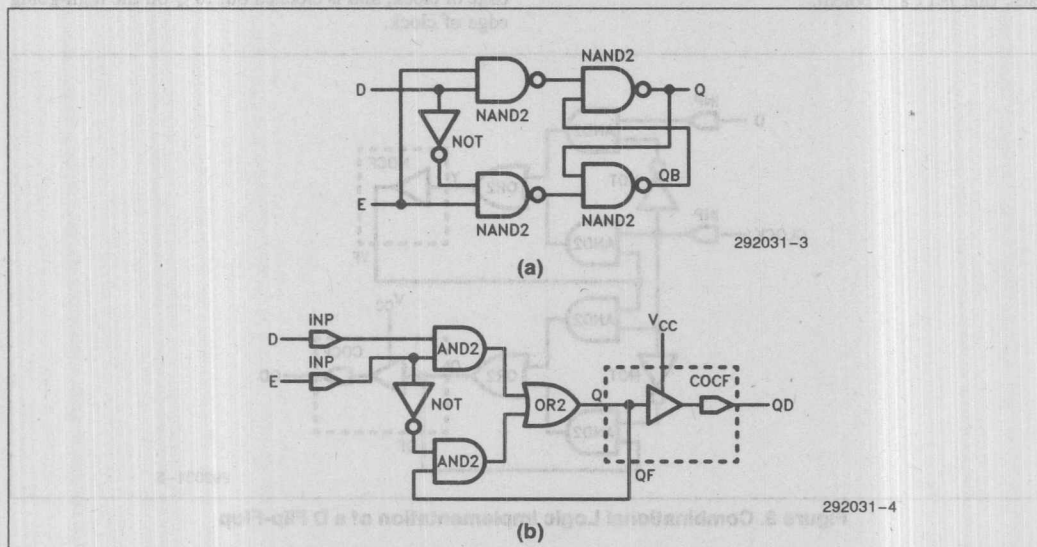


Figure 2. Implementation of a D Type Latch Using a) Discrete Gates and b) EPLD Logic

This latch can be cascaded with a second latch to produce an edge triggered, master/slave D flip-flop, using combinational logic. The flip-flop is a solution to using asynchronous clocking, preset and clear functions when they aren't supported. Also, if an I/O conflict exists within a macrocell group when using registered logic, this design will fit since it uses combinational logic. Figure 3 shows the schematic for this design.

The boolean equation of the D flip-flop is this:

$$QD, QF = COCF(Q, VCC)$$

$$YF = \text{NOCF}(Y)$$

$$Y = D * !CLOCK + YF * CLOCK;$$

$$Q = YF * \text{CLOCK} + QF * !\text{CLOCK};$$

Q is the flip-flop output and Y is the first latch output. Data is latched in to the second latch on the low-going edge of clock, and is clocked out to Q on the high-going edge of clock.

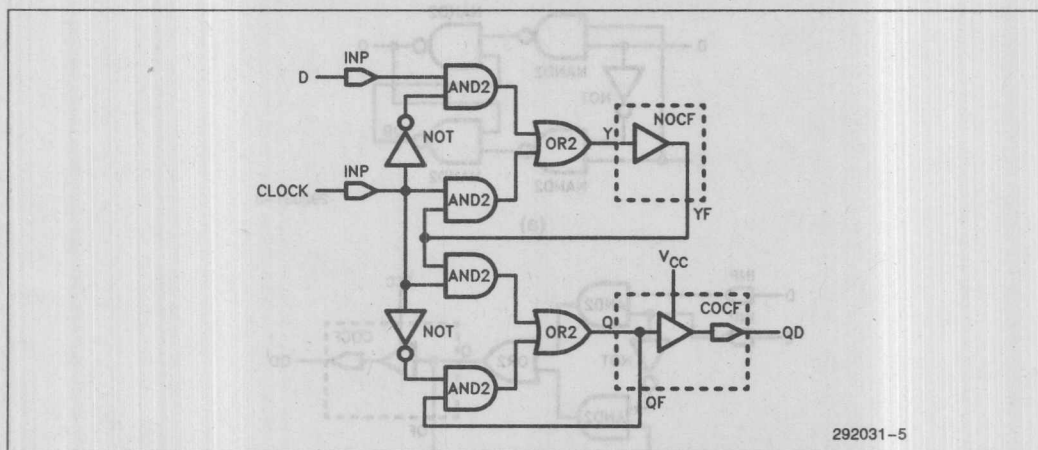


Figure 3. Combinational Logic Implementation of a D Flip-Flop

Preset and clear can be added into the equations as well:

$$QD, QF = COCF(Q, VCC)$$

$$YF = \text{NOCF}(Y)$$

$$Y = D * !CLOCK + YF * CLOCK;$$

$$Q = YF * \text{CLOCK} * !(\text{CLEAR TERM}) + (\text{PRESET TERM}) +$$

QF \* !CLOCK \* ! (CLEAR TERM):

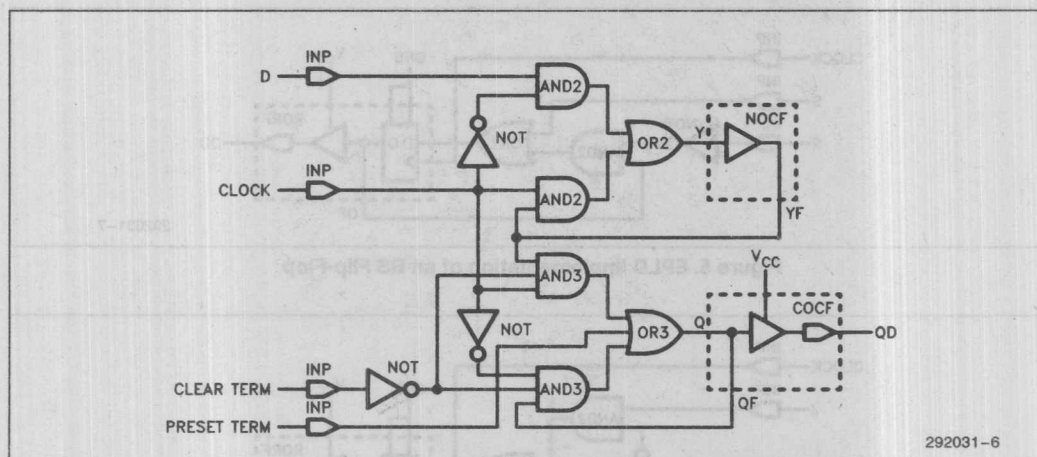
When the PRESET TERM is logically true, Q is asynchronously set to 1.

When the CLEAR TERM is logically true, Q is asynchronously cleared to 0.

The PRESET TERM takes priority over the CLEAR TERM.

This schematic is shown in Figure 4.

Due to the nature of the design, input delays plus array delays plus feedback delays must be added and used to determine a maximum operating frequency. In this example,  $t_{IN} + t_{AD} + t_{CF} + t_{AD} = 113$  ns for a -65 5C121, leaving a maximum frequency of 8.8 MHz.



### Figure 4. D Flip-Flop with Added Preset and Clear Terms



Other useful workarounds involve D registers and logic in constructing RS, JK and T flip-flops, for use in EPLDs not supporting these configurations. The RS flip-flop is simply the RS latch discussed earlier coupled to registered feedback.

$$QD, QF = \text{RORF}(Q, \text{CLOCK}, \text{GND}, \text{GND}, \text{VCC})$$

$$Q = S + QF * !R;$$

Normally, S and R will remain low. When S is brought high, QD will become 1 on the next clock trigger edge. When R is brought high, QD will become 0 on the next clock trigger edge. The schematic is given in Figure 5.

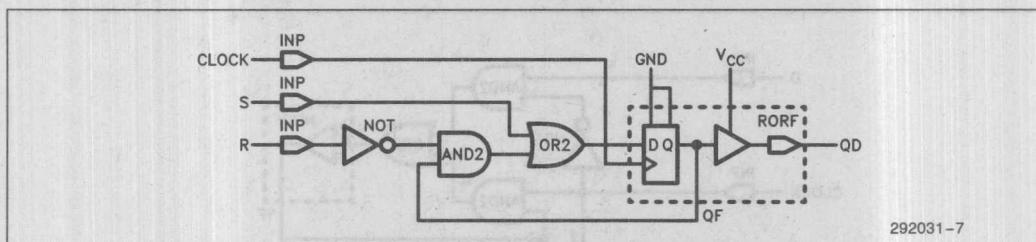


Figure 5. EPLD Implementation of an RS Flip-Flop

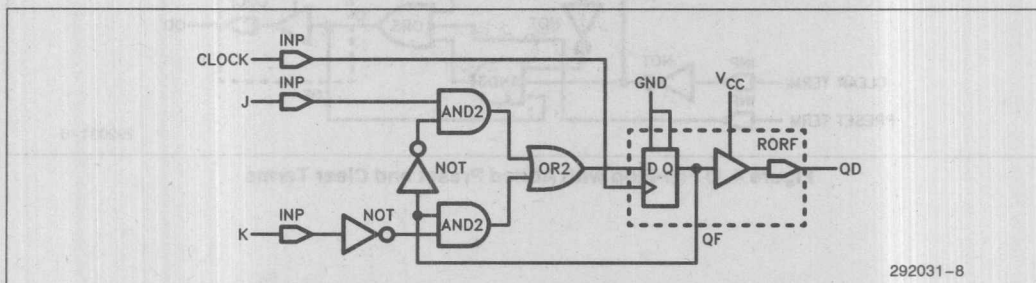


Figure 6. EPLD Implementation of a JK Flip-Flop

The T flip-flop is also easily constructed:

$$QD, QF = RORF(Q, CLOCK, GND, GND, VCC)$$

$$Q = T * !QF + !T * QF;$$

When T is high, QD will toggle to opposite state on next trigger. When T is low, QD will remain the same. Figure 7 shows the T flip-flop design schematic.

Each of these designs uses a minimum number of p-terms; adding p-terms is possible to the limit of the macrocell being used. It is possible to substitute an entire logical expression for each input listed (except

register clock), as long as the minimized logic equations resulting do not exceed the macrocells p-term count.

For example, consider using the J-K register. Setting  $J = A * B * C + D$  and setting  $K = E * !F * !G + H + I$  then the minimized p-term count will expand from two p-terms to five p-terms, which would still be okay within a macrocell with more than five p-terms.

Using logic gates and combinational or registered feedback, one can easily implement many types of latches and registers. Regardless of the EPLD type, there exists the resources to implement any of the discussed circuitry.

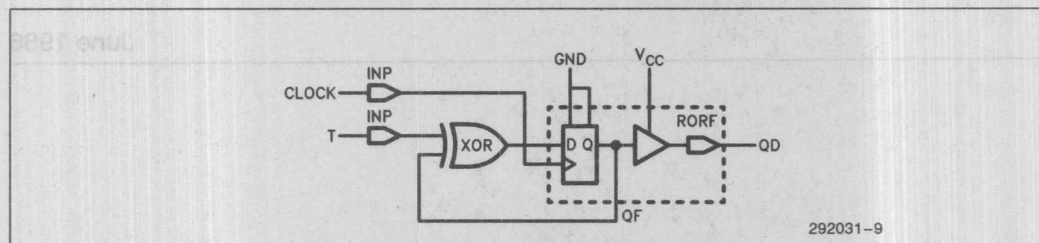


Figure 7. Implementation of a T Flip-Flop

June 1986

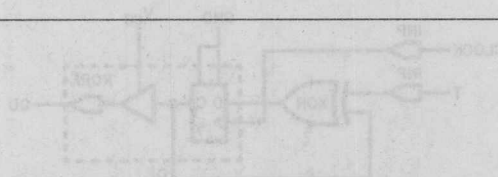


Figure 2. Implementation of a T flip-flop

# The 5C060

## Unification of a CHMOS System

**J. R. DONNELL**  
PROGRAMMABLE LOGIC APPLICATIONS  
INTEL CORPORATION

Order Number: 292009-003

# OF A CHMOS SYSTEM

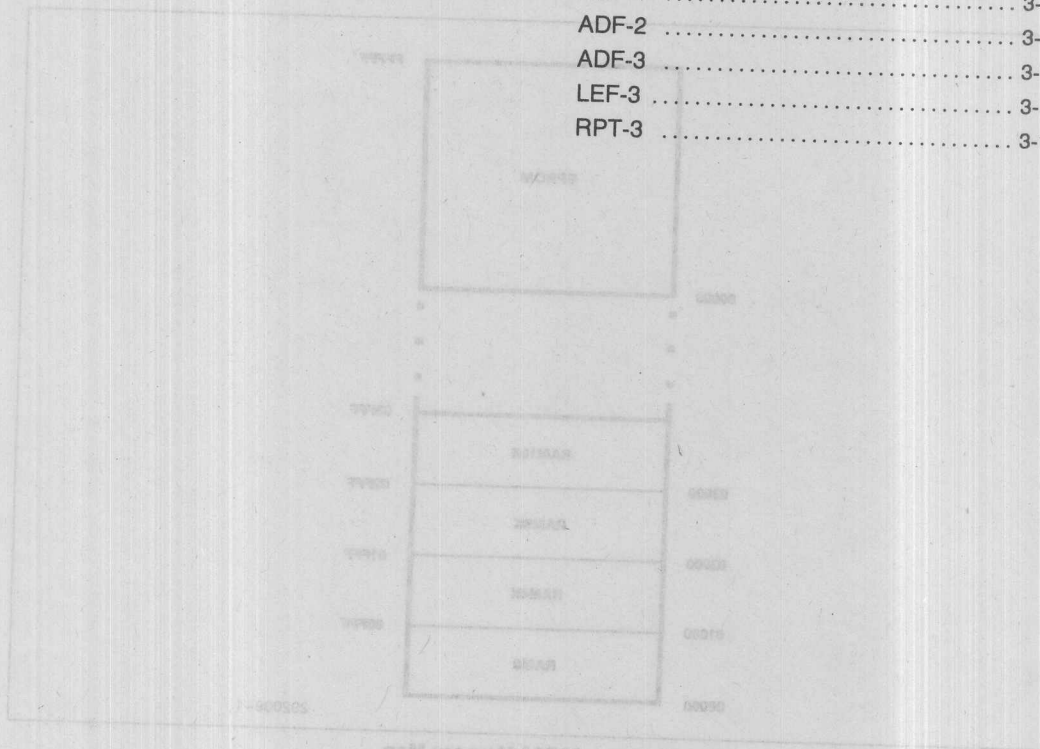
PAGE

<b>INTRODUCTION</b>	3-168
<b>OBJECTIVE</b>	3-168
<b>MEMORY DECODING</b>	3-168
Power Down	3-169
Wait States	3-170
<b>LOC FILES</b>	3-170
JEDEC File	3-170
LEF File	3-170
Utilization Report	3-171
<b>SUMMARY</b>	3-171
<b>APPENDIX</b>	3-172
ADF-1	3-172
ADF-2	3-173
ADF-3	3-174
LEF-3	3-175
RPT-3	3-176

3

The system in question supports one 32K bank of EPROM memory, and four banks of 4K static RAM. Figure 1 shows the memory map of this system. Address lines A19, A18, and A17 will be used to decode the address space. PWR\_DWN and ST\_MIO serve as enable. In addition, to avoid data bus contention signals memory read (MRDC) and advanced memory write (AMWC) are decoded along with the address lines for RAM chip select. This is necessary for devices without output enables (OE) on multiplexed address/data buses.

## MEMORY DECODING





## INTRODUCTION

From an outside glance, the world of computers and microprocessors seems filled with dedicated ICs that fulfill a variety of system needs. Upon closer inspection we find that designers must still reach into their bag of random logic to link together all of the parts of the system. It seems a shame to stuff a board full of high powered peripherals and still have portions of that board wasted on decoders, latches, and other miscellaneous random logic.

True, programmable logic has been around a long time. But that logic is somewhat rigid in form, one time programmable, and can also double as space heaters. These devices are totally unacceptable for a CMOS system. What is needed is a flexible PLA architecture, erasability for prototyping, and CMOS for low power. In addition, for this particular application the device must perform from static operation to 10 MHz.

## OBJECTIVE

This application note covers the design of three separate circuits for Intel's CHMOS Design Kit. The functions performed by the 5C060 are: Memory decoding, wait state generation, and the power down circuitry for the 80C88 system clock.

## MEMORY DECODING

The system in question supports one 32K bank of EPROM memory, and four banks of 4K static RAM. Figure 1 shows the memory map of this system. Address lines A19, A13, and A12 will be used to decode the address space. PWR\_DWN and S2\_MIO serve as enables. In addition, to avoid data bus contention signals memory read (MRDC) and advanced memory write (AMWC) are decoded along with the address lines for RAM chip selects. This is necessary for devices without output enables (OE) on multiplexed address/data busses.

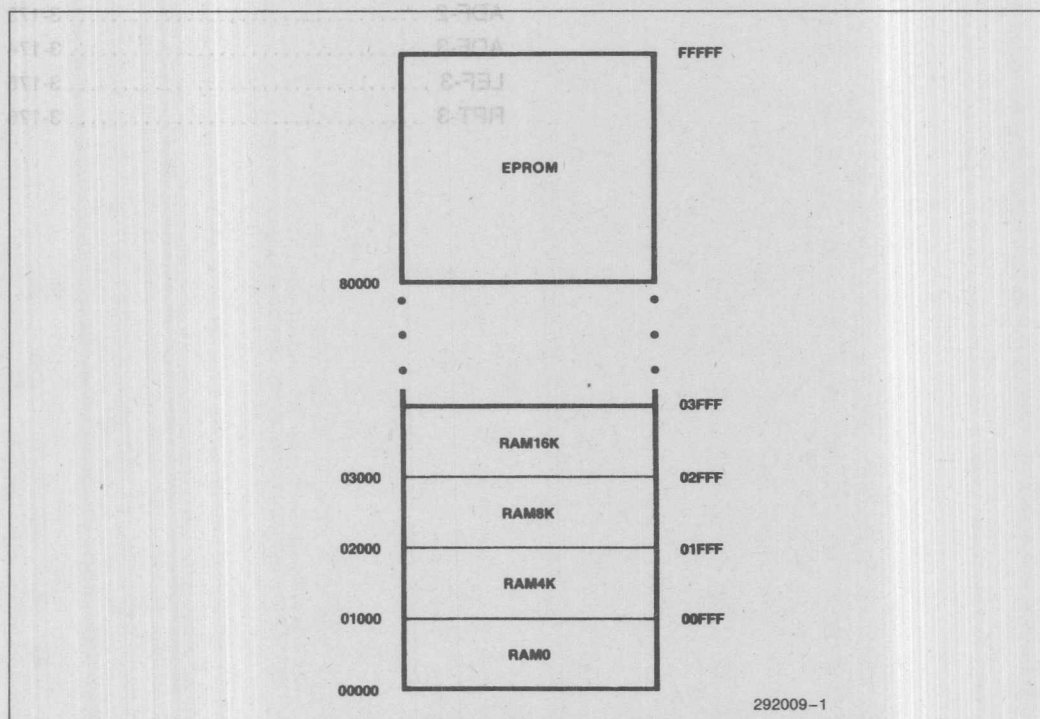


Figure 1. 80C88 Memory Map

Figure 2 shows a discrete implementation of the chip select decoding logic.

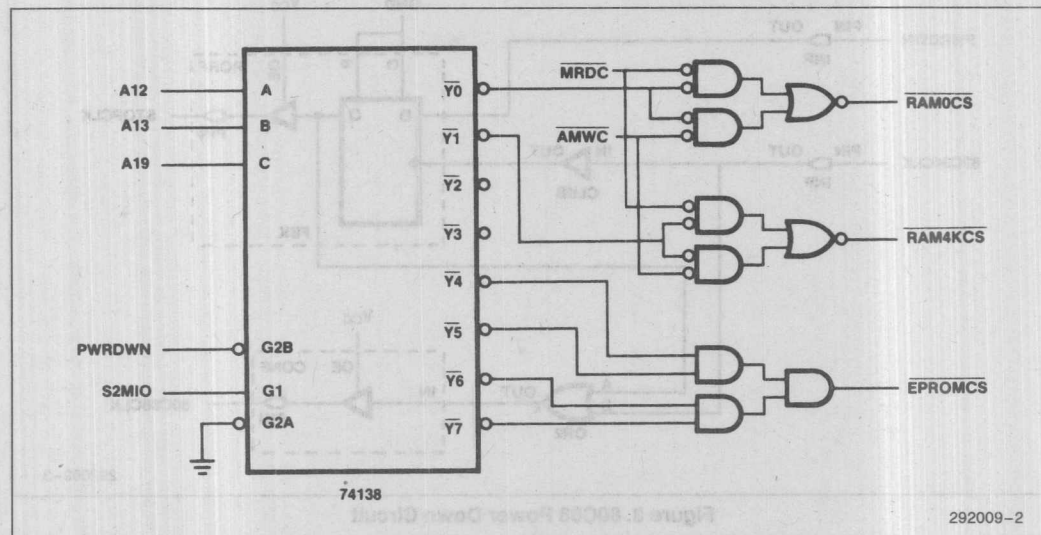


Figure 2. Discrete Decoding Logic Solution

Several options for entering this design are available through Intel's Programmable Logic Development System (iPLDS). (For a more complete description of iPLDS the reader is referred to the iPLDS data sheet.) The design entry vehicle chosen for this application note is the Logic Builder. (Logic Builder is an interactive netlist method of design entry especially suited to Boolean equation entry and entry from existing schematics.) Several reasons are behind this decision. First, the Logic Builder software is included in iPLDS. In addition, Logic Builder entry is very fast, the designer may choose either netlist entry or Boolean equations, and finally, the Logic Builder software makes additions and corrections of design very easy.

Using Logic Builder, the first step for this design is to determine the equations for the 3 to 8 decoder shown in Figure 2. These equations are simply the decoding of the address lines ANDed with the enable signal. Equations 0 thru 8 implement the decoding function of Figure 2.

```

/Y0 = /A19*/A13*/A12*ENABLE; (0)
/Y1 = /A19*/A13*/A12*ENABLE; (1)
/Y2 = /A19*/A13*/A12*ENABLE; (2)
/Y3 = /A19*/A13*/A12*ENABLE; (3)
/Y4 = /A19*/A13*/A12*ENABLE; (4)
/Y5 = /A19*/A13*/A12*ENABLE; (5)
/Y6 = /A19*/A13*/A12*ENABLE; (6)
/Y7 = /A19*/A13*/A12*ENABLE; (7)
ENABLE = /PWRDWN*S2MIO; (8)

```

Armed with this knowledge it becomes trivial to enter the circuit of Figure 2 into Logic Builder. Included in the Appendix is the Advanced Design File (ADF) created by Logic Builder for this circuit (ADF-1). Typically the ADF would now be submitted to the Logic Optimizing Compiler (LOC) for Boolean minimization and design fitting. In this case we have used only a small portion of the logic available in the 5C060 so let us continue with the wait state generator and power down circuitry.

### Power Down

Since this design is based on the 80C88 we can actually stop the system clock for extended periods of time and power back up as if nothing had occurred. The circuit to achieve this power down is shown in Figure 3.

As long as the PWRDWN signal is low the 82C84 clock output is OR'ed with a logical zero from the PWRDWN flip-flop. As a result the 82C84 drives the 80C88 system clock. If PWRDWN goes HIGH, the rising edge of the next 82C84 clock will set the output of the PWRDWN flip-flop HIGH inhibiting the fall of the next clock cycle. The 80C88 system clock will remain HIGH until PWRDWN goes LOW and the PWRDWN flip-flop is clocked from the 82C84 clock. Using this configuration we avoid partial clock cycles for the 80C88 system clock.

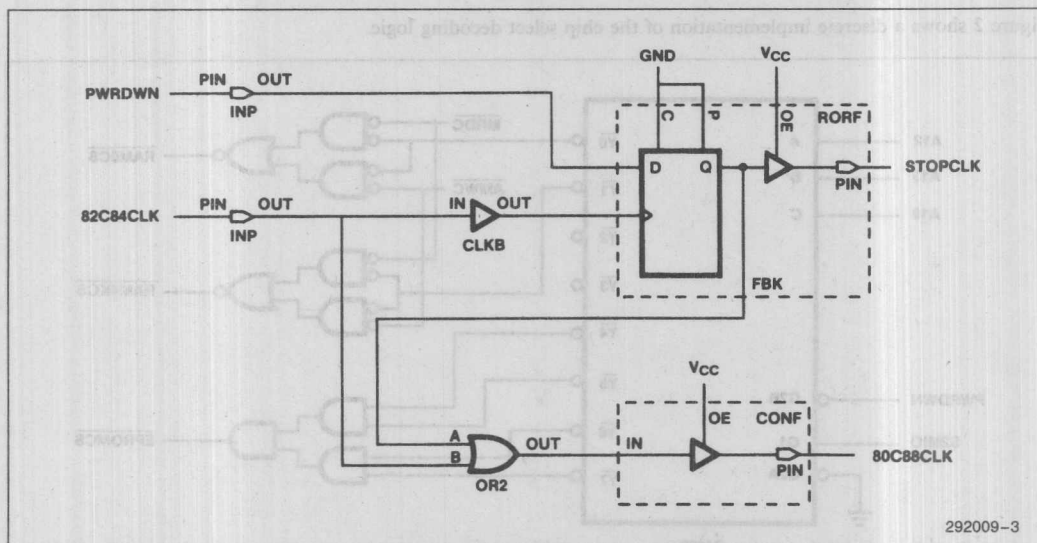


Figure 3. 80C88 Power Down Circuit

Again, entering this circuit into Logic Builder is trivial. In fact it can be added directly to the decoder circuit shown above. The ADF file for this addition is shown in the appendix under ADF-2.

### Wait States

The majority of memory and peripheral devices which fail to operate at the maximum CPU frequency typically do not require more than one wait state. The circuit shown in Figure 4 is an example of a simple wait state generator. The circuit operation is as follows. Given that a memory location requiring a wait state has been selected, ALE in conjunction with /WAITCS will clear the flip-flop—driving the 82C84RDY line high low. The 82C84 samples the RDY line during T2 of the 80C88 bus cycle, and in this case detects a wait state. The rising edge of T2 then clocks the 82C84RDY line high thereby inserting only one wait state.

Once again, adding this circuit to the existing decoder and power down design is simple. The final ADF file is given in the appendix under ADF-3. Once the final design has been completed the ADF is submitted to the Logic Optimizing Compiler. LOC compiles the design, performs Boolean minimization, and fits the design into the target EPLD. In addition, LOC produces two files. The JEDEC programming file, the Logic Equation File

(LEF), and the Utilization Report. These are also included in the appendix for each step in this design process.

### LOC FILES

#### The JEDEC File

The JEDEC file is analogous to the object code file that is used to program EPROMs. This file is used by the Logic Programming Software (LPS) to program Intel's EPLDs.

#### The LEF File

The LEF file is an optional file produced by the compiler. The LEF file contains the minimized Boolean equations which resulted from the original ADF. Some interesting points can be raised concerning the LEF file. Looking at LEF-3, first recall that the EPROM chip select was a function of A19, A13, A12, and the enable signals. It turns out that after minimization the EPROM chip select depends only on A19 and the enable signals (/PWRDWN and S2MIO). This is shown in the LEF file. One other point, the initial wait state circuitry employed a JK flip-flop. The compiler automatically minimized this circuit into a D-type flip-flop with feedback achieving the same functionality.

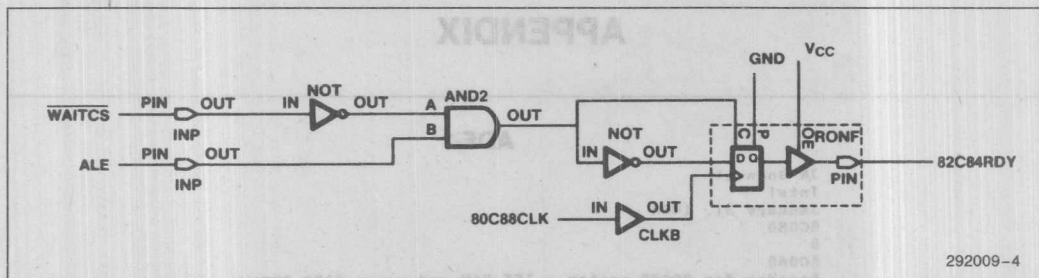


Figure 4. Single Wait State Generator for the 80C88

## The Utilization Report

Finally, the Utilization Report contains the pin-out for the design, information about the architectural layout of the design, and a percent utilization for pins, macrocells, and product terms. Examining the utilization report for this design we find that two of the sixteen macrocells are still available. We could therefore add more functionality in the same 24 pin package. Possible additions would be more memory decoding, invalid memory detection, additional wait state generators, etc. One point should be raised: The circuitry designed in this applications note is relatively simple compared to the complex logic functions that could be implemented in the 5C060.

## SUMMARY

The designs shown in this applications note are typical requirements of any microprocessor system. The 5C060 provided a single chip solution to bind together the primary elements of that system. Few other types of programmable logic could implement the same logic in a single package. None could do it in CMOS erasable logic. The 5C060 has room for more.



# APPENDIX

## ADF-1

JR Donnell

Intel

January 31, 1986

5C060

0

5C060

Decoder for 80C88 system - 16K RAM and upper 512K EPROM

IR Version 3.0, Release 17x, 9/26/85

PART: 5C060

INPUTS: A19, A13, A12, PWRDWN, S2MIO, AMWC, MRDC

OUTPUTS: RAM0CS, RAM4KCS, RAM8KCS, RAM16KCS, RPROMCS

NETWORK:

RAM0CS = CONF (RAM0CS, VCC)

RAM4KCS = CONF (RAM4KCS, VCC)

RAM8KCS = CONF (RAM8KCS, VCC)

RAM16KCS = CONF (RAM16KCS, VCC)

RPROMCS = CONF (EPROMCS, VCC)

A19 = INP (A19)

A13 = INP (A13)

A12 = INP (A12)

PWRDWN = INP (PWRDWN)

S2MIO = INP (S2MIO)

MRDC = INP (MRDC)

AMWC = INP (AMWC)

EQUATIONS:

RAM8KCS = ((/MRDC\*Y2

+ /AMWC\*Y2):

RAM16KCS = ((/MRDC\*Y3

+ /AMWC\*Y3):

RPROMCS = ((/Y7

+ /Y6

+ /Y5

+ /Y4):

Y7 = ((A19\*A13\*A12\*RNARL):

Y6 = ((A19\*A13\*/A12\*ENABLE):

Y5 = ((A19\*/A13\*A12\*RNARL):

Y4 = ((A19\*/A13\*/A12\*ENABLE):

RNARL = /PWRDWN\*S2MIO:

Y3 = ((/A19\*A13\*A12\*ENABLE):

Y2 = ((/A19\*A13\*/A12\*RNARL):

RAM4KCS = ((/MRDC\*Y1

+ /AMWC\*Y1):

Y1 = ((/A19\*/A13\*A12\*ENABLE):

RAM0CS = ((/MRDC\*Y0

+ /AMWC\*Y0):

Y0 = ((/A19\*/A13\*/A12\*RNARL):

RND\$

292009-5

# ADF-2

JR Donnell

Intel

January 31, 1986

5C060

0

5C060

Decoder for 80C88 system - 16K RAM and upper 512K EPROM  
Plus power down circuit

LB Version 3.0, Baseline 17x, 9/26/85

PART: 5C060

INPUTS: A19,A13,A12,PWRDWN,S2MIO,AMWC,MRDC,R2C84CLK

OUTPUTS: RAM0CS,RAM4KCS,RAM8KCS,RAM16KCS,RPROMCS,STOPCLK,R0C88CLK

NETWORK:

RAM0CS = CONF (RAM0CS,VCC)

RAM4KCS = CONF (RAM4KCS,VCC)

RAM8KCS = CONF (RAM8KCS,VCC)

RAM16KCS = CONF (RAM16KCS,VCC)

RPROMCS = CONF (EPROMCS,VCC)

STOPCLK,STOPCLKF = RORF (PWRDWN,R2C84CLKR,GND,GND,VCC)

R0C88CLK = CONF (R0C88CLK,VCC)

PWRDWN = INP (PWRDWN)

R2C84CLKB = CLKB (R2C84CLK)

R0C88CLK = OR (STOPCLKF,R2C84CLK)

R2C84CLK = INP (R2C84CLK)

A19 = INP (A19)

A13 = INP (A13)

A12 = INP (A12)

S2MIO = INP (S2MIO)

MRDC = INP (MRDC)

AMWC = INP (AMWC)

EQUATIONS:

RAM0CS =  $((MRDC \cdot Y0) +$

$AMWC \cdot Y0):$

RAM4KCS =  $((MRDC \cdot Y1) +$

$AMWC \cdot Y1):$

RAM8KCS =  $((MRDC \cdot Y2) +$

$AMWC \cdot Y2):$

RAM16KCS =  $((MRDC \cdot Y3) +$

$AMWC \cdot Y3):$

RPROMCS =  $((Y7) +$

$Y6$

$+ Y5$

$+ Y4):$

Y0 =  $((A19 \cdot A13 \cdot A12 \cdot \text{ENABLE}):$

$Y1 = ((A19 \cdot A13 \cdot A12 \cdot \text{ENABLE}):$

$Y2 = ((A19 \cdot A13 \cdot A12 \cdot \text{ENABLE}):$

$Y3 = ((A19 \cdot A13 \cdot A12 \cdot \text{ENABLE}):$

$Y7 = ((A19 \cdot A13 \cdot A12 \cdot \text{ENABLE}):$

$Y6 = ((A19 \cdot A13 \cdot A12 \cdot \text{ENABLE}):$

$Y5 = ((A19 \cdot A13 \cdot A12 \cdot \text{ENABLE}):$

$Y4 = ((A19 \cdot A13 \cdot A12 \cdot \text{ENABLE}):$

ENABLE = PWRDWN \cdot S2MIO:

RND\$

JR Donnell  
Intel  
January 31, 1986  
5C060  
0

## ADF-3

5C060  
Decoder for 80C88 system - 16K RAM and upper 512K EPROM  
Plus power down circuit  
Plus wait state circuit  
LR Version 3.0, Baseline 17x, 9/26/85  
PART: 5C060  
INPUTS: A19, A13, A12, PWRDWN, S2MIO, AMWC, MRDC, R2C84CLK, ALE, WAITCS  
OUTPUTS: RAM0CS, RAM4KCS, RAM8KCS, RAM16KCS, EPROMCS, STOPCLK, R0C88CLK, R2C84RDY  
NETWORK:  
RAM0CS = CONF (RAM0CS, VCC)  
RAM4KCS = CONF (RAM4KCS, VCC)  
RAM8KCS = CONF (RAM8KCS, VCC)  
RAM16KCS = CONF (RAM16KCS, VCC)  
EPROMCS = CONF (EPROMCS, VCC)  
STOPCLK, STOPCLKF = RORF (PWRDWN, R2C84CLK, GND, GND, VCC)  
R0C88CLK, R0C88CLKF = COIF (R0C88CLK, VCC)  
R2C84RDY = RORF (R2C84RDYD, R0C88CLK, R2C84RDYD, GND, VCC)  
PWRDWN = INP (PWRDWN)  
R2C84CLK = CLKR (R2C84CLK)  
R0C88CLK = OR (STOPCLKF, R2C84CLK)  
R2C84CLK = INP (R2C84CLK)  
A19 = INP (A19)  
A13 = INP (A13)  
A12 = INP (A12)  
S2MIO = INP (S2MIO)  
MRDC = INP (MRDC)  
AMWC = INP (AMWC)  
R0C88CLKB = CLKB (R0C88CLKF)  
WAITCS = INP (WAITCS)  
ALE = INP (ALE)  
ROUTATIONS:  
RAM0CS = /(MRDC\*Y0  
+ /AMWC\*Y0);  
RAM4KCS = /(MRDC\*Y1  
+ /AMWC\*Y1);  
RAM8KCS = /(MRDC\*Y2  
+ /AMWC\*Y2);  
RAM16KCS = /(MRDC\*Y3  
+ /AMWC\*Y3);  
EPROMCS = /(Y7  
+ /Y5  
+ /Y4);  
Y0 = /(A19\*/A13\*/A12\*ENABLE);  
Y1 = /(A19\*/A13\*/A12\*RNABLR);  
Y2 = /(A19\*/A13\*/A12\*ENABLE);  
Y3 = /(A19\*/A13\*/A12\*RNABLR);  
Y7 = /(A19\*/A13\*/A12\*ENABLE);  
Y6 = /(A19\*/A13\*/A12\*RNABLR);  
Y5 = /(A19\*/A13\*/A12\*ENABLE);  
Y4 = /(A19\*/A13\*/A12\*RNABLR);  
ENABLE = /PWRDWN\*S2MIO;  
R2C84RDYD = /R2C84RDY;  
R2C84RDY = /WAITCS\*ALE;  
RND\$

292009-7

JR Donnell  
Intel  
January 31, 1986  
5C060  
0  
5C060

Decoder for 80C88 system - 16K RAM and upper 512K EPROM  
Plus power down circuit  
Plus wait state circuit  
IR Version 3.0, Baseline 17x, 9/26/85

PART:

5C060

INPUTS:

A19, A13, A12, PWRDWN, S2MIO, AMWC, MRDC, R2C84CLK, ALE, WAITCS

OUTPUTS:

RAM0CS, RAM4KCS, RAM8KCS, RAM16KCS, EPROMCS, STOPCLK, R2C88CLK,  
R2C84RDY

NETWORK:

A19 = INP(A19)  
A13 = INP(A13)  
A12 = INP(A12)  
PWRDWN = INP(PWRDWN)  
S2MIO = INP(S2MIO)  
AMWC = INP(AMWC)  
MRDC = INP(MRDC)  
R2C84CLK = INP(R2C84CLK)  
ALE = INP(ALE)  
WAITCS = INP(WAITCS)  
RAM0CS = CONF(RAM0CS, VCC)  
RAM4KCS = CONF(RAM4KCS, VCC)  
RAM8KCS = CONF(RAM8KCS, VCC)  
RAM16KCS = CONF(RAM16KCS, VCC)  
EPROMCS = CONF(EPROMCS, VCC)  
..SG000D = CLKB(R2C84CLKB)  
STOPCLK, STOPCLKF = RORF(PWRDWN, ..SG000D, GND, GND, VCC)  
R2C88CLK, R2C88CLKF = COIF(R2C88CLK, VCC)  
..SG001D = CLKB(R2C88CLKB)  
R2C84RDY = RORF(R2C84RDYD, ..SG001D, R2C84RDYC, GND, VCC)

EQUATIONS:

R2C84RDYC = WAITCS' \* ALE;  
..SG001D = R2C88CLKF;  
R2C84RDYD = (WAITCS' \* ALE)';  
R2C88CLK = (STOPCLKF' \* R2C84CLK')';  
..SG000D = R2C84CLK;  
EPROMCS = (A19 \* PWRDWN' \* S2MIO)';  
RAM16KCS = MRDC \* AMWC  
+ A19' \* A13 \* A12 \* PWRDWN' \* S2MIO;  
RAM8KCS = MRDC \* AMWC  
+ A19' \* A13 \* A12' \* PWRDWN' \* S2MIO;  
RAM4KCS = MRDC \* AMWC  
+ A19' \* A13' \* A12 \* PWRDWN' \* S2MIO;  
RAM0CS = MRDC \* AMWC  
+ A19' \* A13' \* A12' \* PWRDWN' \* S2MIO;

RND\$

292009-8



## RPT-3

## Logic Optimizing Compiler Utilization Report

\*\*\*\* Design implemented successfully

JR Donnell

Intel

January 31, 1986

5C060

0

5C060

Decoder for 80C88 system - 16K RAM and upper 512K EPROM

Plus power down circuit

Plus wait state circuit

L.R. Version 3.0. Baseline 17x. 9/26/85

## 5C060

```

GND - 1 24:- Vcc
PWRDWN - 2 23:- A19
GND - 3 22:- STOPCLK
GND - 4 21:- R2CR4RDY
WAITCS - 5 20:- 80C88CLK
A1R - 6 19:- RPRMCS
R2CR4CLK - 7 18:- RAM16KCS
MRDC - 8 17:- RAMRKCS
AMWC - 9 16:- RAM4KCS
S2MTO - 10 15:- RAM0CS
A12 - 11 14:- A13
GND - 12 13:- GND

```

\*\*TNP/ITS\*\*

```

Name Pin Resource MCell #0 PTerms : MCells Feeds:
PWRDWN 2 OVC INP 11 0/ 8 2 1 4 5 6 7 8
WAITCS 5 TNP 11 0/ 8 2 - -2 -
ALE 6 INP 12 0/ 8 2 - 2 -
R2CR4CLK 7 TNP 13 0/ 8 3 - - 1
MRDC 8 INP 14 0/ 8 5 - - -
AMWC 9 OVC INP 15 0/ 8 5 6 7 8
S2MTO 10 TNP 16 0/ 8 4 5

```

292009-9

A12 11 TNP

-

-

6  
7  
8

5

-

-

-

A13 14 TNP

-

-

5

-

-

-

A19 23 TNP

-

-

4

-

-

-

5

6

7

8

# \*\*OUTPUTS\*\*

Name	Pin	Resource	MCell #	PTerms	MCells	Feeds: OR	Clear	Clock
RAM0CS	15	CONF	8	2/ 8	-	-	-	-
RAM4KCS	16	CONF	7	2/ 8	-	-	-	-
RAM8KCS	17	CONF	6	2/ 8	-	-	-	-
RAM16KCS	18	CONF	5	2/ 8	-	-	-	-
RPPROMCS	19	CONF	4	1/ 8	-	-	-	-
R0CRRCLK	20	COTF	3	1/ 8	-	-	-	2
R2C84RDY	21	R0NFA	2	1/ 8	-	-	-	-
STOPCLK	22	R0RFA	1	1/ 8	3	-	-	-

# \*\*UNUSED RESOURCES\*\*

Name	Pin	Resource	MCell	PTerms
-	1	-	-	-
-	3	-	9	8
-	4	-	10	8
-	13	-	-	-

# \*\*PART UTILIZATION\*\*

81% Pins  
87% MacroCells  
9% Pterms

292009-10

June 1986

# Implementing a CMOS Bus Arbiter/Controller in the 5C060 EPLD

**DANIEL E. SMITH**  
APPLICATIONS ENGINEERING  
INTEL CORPORATION

Order Number: 292012-001

## BUS ARBITER/ CONTROLLER IN THE 5C060 EPLD

## WOLFGANG PAGE

## 3

3-179



## INTRODUCTION

This application note shows how to implement a CMOS Bus Arbiter/Controller in an Intel 5C060 EPLD (Erasable Programmable Logic Device). The note includes a brief overview of a similar circuit implemented with typical PLA devices, a more detailed discussion of the 5C060 implementation, and a summary.

The bus priority resolution and arbitration scheme selected for the circuit is that used by the industry-standard MULTIBUS I interface. Operation and timing for the MULTIBUS I interface is well understood by most engineers and is described in readily available Intel publications. Thus, a description of the MULTIBUS I interface is not included here. The bus arbiter/controller functions shown here support both serial and parallel priority resolution between bus masters. Timing is equivalent to MULTIBUS I specifications. Electrical specifications for both the PLA and EPLD approaches vary from MULTIBUS I standards. Neither of the two circuits discussed here provide the full current sink capability for all MULTIBUS I signals. Because the EPLD implementation is designed for CMOS systems, however, this requirement is not relevant for the 5C060 implementation.

## PLA APPROACH

The functional equivalent of a MULTIBUS I arbiter/controller can be implemented in two 20-pin PLA-type devices as shown in Figures 1 and 2. (Figure 1 shows the logic for the arbiter device. Figure 2 shows the logic for the controller and the connections to the arbiter.) Figure 3 shows the arbiter list file as an example of PLA-type files. Two different 20-pin PLA devices are required to implement the arbiter and controller functions, a 16R4-type device and a 16L8-type device.

Implementation of logic devices in PLA-type devices, such as those shown here, has proven to be quite beneficial. Development time and cost is much less than for custom silicon device designs. The two PLA-type devices take up less board space than a discrete TTL implementation of the same functions. In addition, the two raw devices can also be used for different functions in other products, thereby reducing inventory costs. As a result of these factors (and others), use of PLA-type devices has grown substantially in recent years.

With the increased density and flexibility of EPLD devices over typical PLA-type devices, even greater space, inventory, and cost savings can be obtained by using EPLDs. The following section shows an implementation of the same arbiter/controller functions in a single 24-pin 5C060 EPLD device.

## 5C060 IMPLEMENTATION

The equivalent functions for both the MULTIBUS I arbiter and controller fit inside a single 5C060 EPLD device. The 5C060 device is available in a 24-pin 0.3" DIP package. Figures 4 and 5 show logic diagrams for the arbiter and controller functions. When compared with the PLA implementation, some differences in the design are immediately apparent. These differences result from the characteristics of the EPLD macrocell or from corrections to the circuit used in Figures 1 and 2.

The major change resulting from the EPLD macrocell structure concerns the EPLD output buffers. Since output buffers from macrocells are non-inverting (PLA-type devices typically contain inverting buffers), signals enter the buffers in the same logic orientation from which they are to appear at the output. The logic for the EPLD (shown in Figures 4 and 5) incorporates this change.

Some errors in the PLA-type implementation have also been corrected in the EPLD design. These changes are as follows:

- The  $M/\overline{IO}$  input to the MRDC/ and MWTC/ gates is inverted.  $M/\overline{IO}$  distinguishes between memory and I/O cycles. The PLA-type implementation does not use this signal properly; the PLA-type controller generates read or write commands to both memory and I/O at the same time, which can result in contention between memory and I/O during bus transfers.
- BPRO/ is gated by BPRN/ in the EPLD design. When using serial priority resolution, this allows the highest priority arbiter to prevent all other masters from controlling the bus. (In the PLA design, BPRO/ is enabled/disabled only by a local request. Higher priority arbiters cannot disable all other arbiters. This can result in contention between bus masters. By gating BPRO/ with BPRN/ in the EPLD design, this source of bus contention is prevented.)

Figure 6 shows the list file for the arbiter/controller device. Figure 7 shows the report file produced by the iPLDS software. This file contains a pinout diagram of the final programmed device and provides a resource usage map for the device.

Most of the input and output signals are self-explanatory to those familiar with Intel processors and the MULTIBUS I interface. The XREQ input is the bus transfer request signal from the address decode logic. The BUSY/ and CBRQ/ outputs are bi-directional, simulated open-collector outputs. These outputs use the iPLDS 5C060 (Combinational-Output I/O-Feedback) primitive in the list file. The BUSY/ signal serves to illustrate this use of EPLD outputs.

A pull-up resistor is used externally (i.e., on the back-plane) to hold  $BUSY/\overline{}$  high when no arbiter is in control of the bus. When the arbiter is granted control of the bus,  $AEN$  is clocked high, which enables the output of the  $BUSY/\overline{}$  driver. Since the input to the  $BUSY/\overline{}$  driver is low during normal operation ( $RESET/\overline{}$  inverted), the enabled driver pulls  $BUSY/\overline{}$  low to signal other arbiters that the bus is in use. When the arbiter is finished using the bus,  $AEN$  goes low to disable the  $BUSY/\overline{}$  driver (three-state output). The pull-up resistor pulls  $BUSY/\overline{}$  high to signal other arbiters that the bus is free for use if needed.

Note that  $BUSY/\overline{}$  is also routed into the bus grant logic as input  $BSI$ .  $BSI$  prevents the arbiter from taking control of the bus (and driving  $BUSY/\overline{}$  low) when some other arbiter already has control of the bus. Thus only one arbiter may pull  $BUSY/\overline{}$  low at any one time.

The one difference between standard MULTIBUS I logic levels and the EPLD implementation described here relates to the  $BCLK/\overline{}$  signal. MULTIBUS I bus arbitration uses the negative-going edge of  $BCLK/\overline{}$  to synchronize events. All 5C060 flip-flops, however, clock on the positive-going edge of  $BCLK/\overline{}$ . If all bus masters in the system use the same arbiter implementation, this poses no problem. Otherwise, an external inverter is required for the  $BCLK/\overline{}$  input.

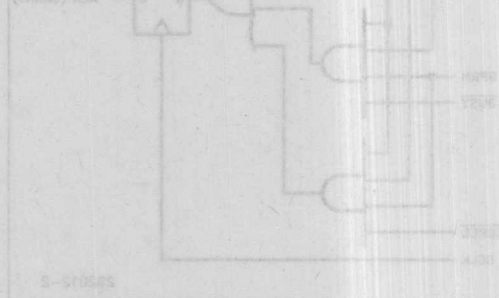


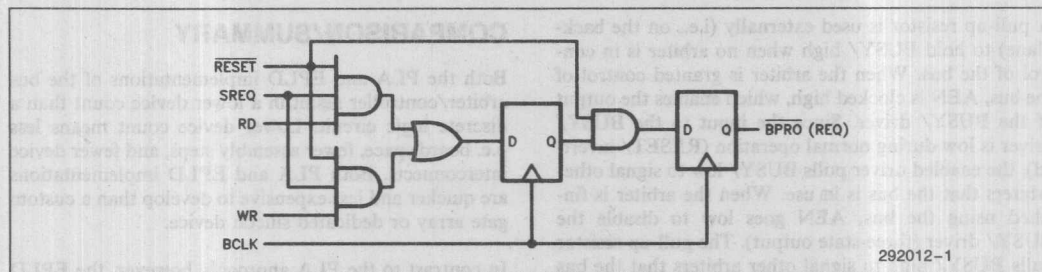
## COMPARISON/SUMMARY

Both the PLA and EPLD implementations of the bus arbiter/controller result in a lower device count than a discrete logic circuit. Lower device count means less p.c. board space, fewer assembly steps, and fewer device interconnects. Both PLA and EPLD implementations are quicker and less expensive to develop than a custom gate array or dedicated silicon device.

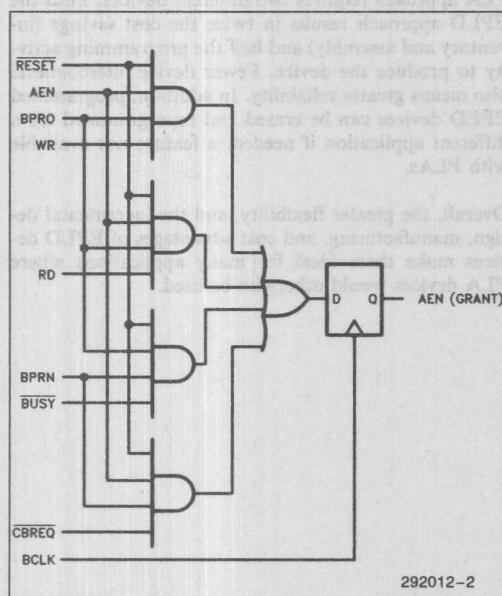
In contrast to the PLA approach, however, the EPLD implementation requires only a single device, while the PLA approach requires two different devices. Thus the EPLD approach results in twice the cost savings (inventory and assembly) and half the programming activity to produce the device. Fewer device interconnects also means greater reliability. In addition, programmed EPLD devices can be erased and reprogrammed for a different application if needed, a feature not available with PLAs.

Overall, the greater flexibility, and the incremental design, manufacturing, and cost advantages of EPLD devices make them ideal for many applications where PLA devices would otherwise be used.

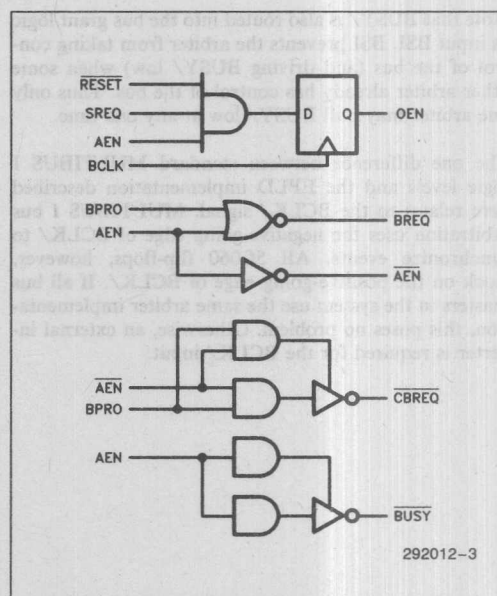




A) Request Synchronizer

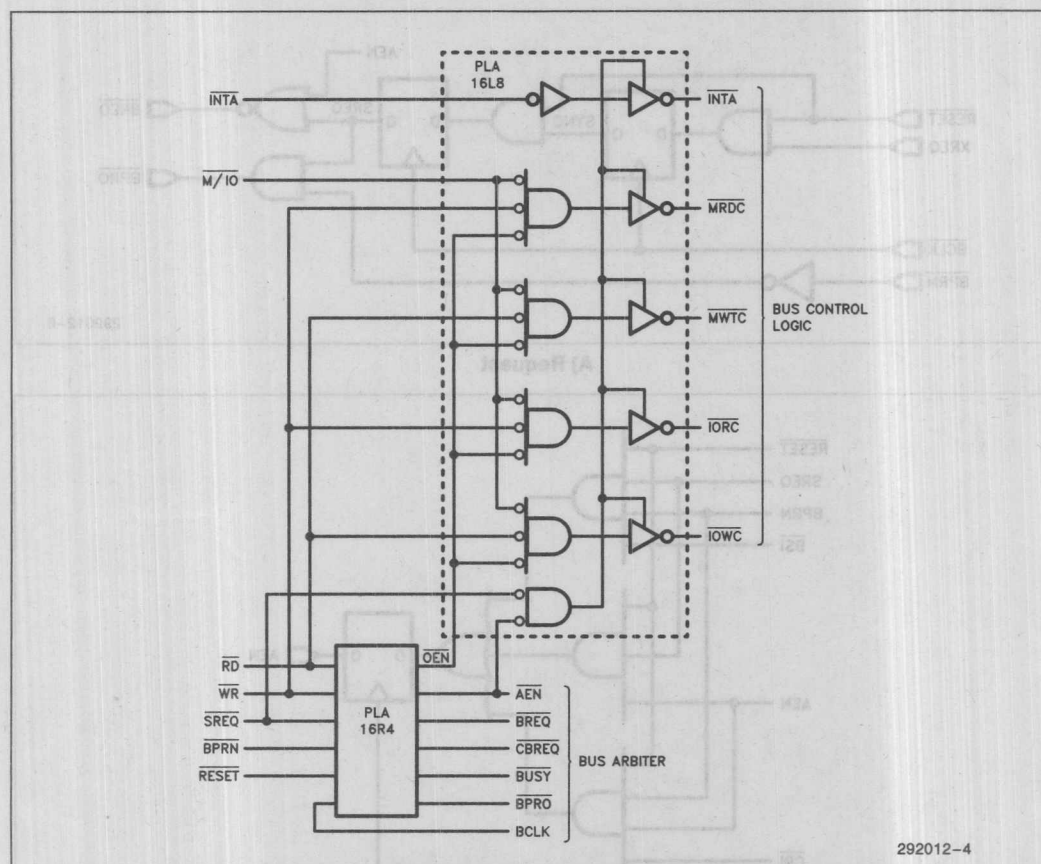


B) Grant/Access Logic



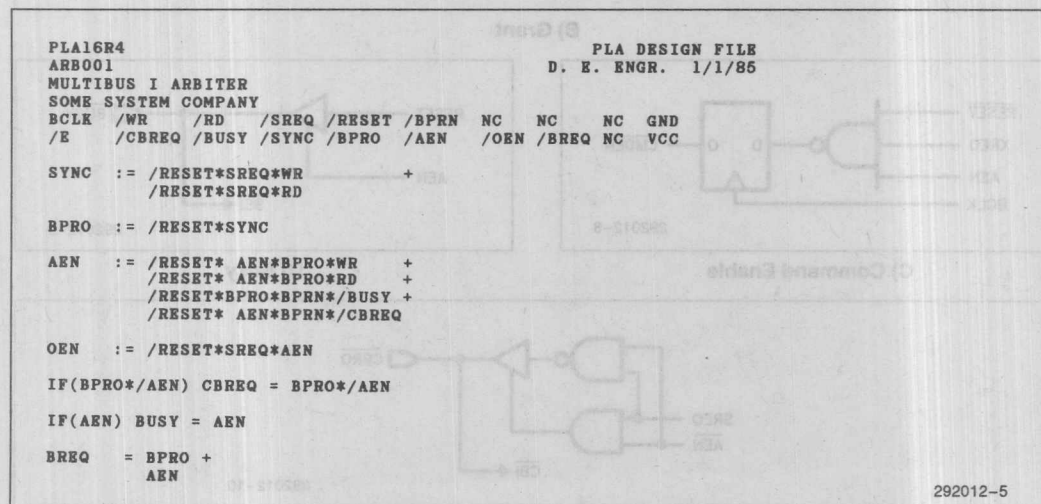
C) Bus Transfer Control

Figure 1. PLA Approach to a Bus Arbiter



292012-4

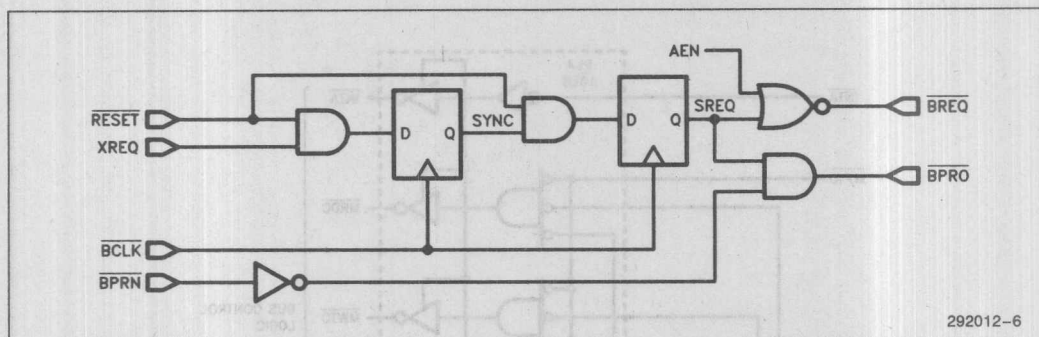
Figure 2. Bus Controller with Arbiter Connected



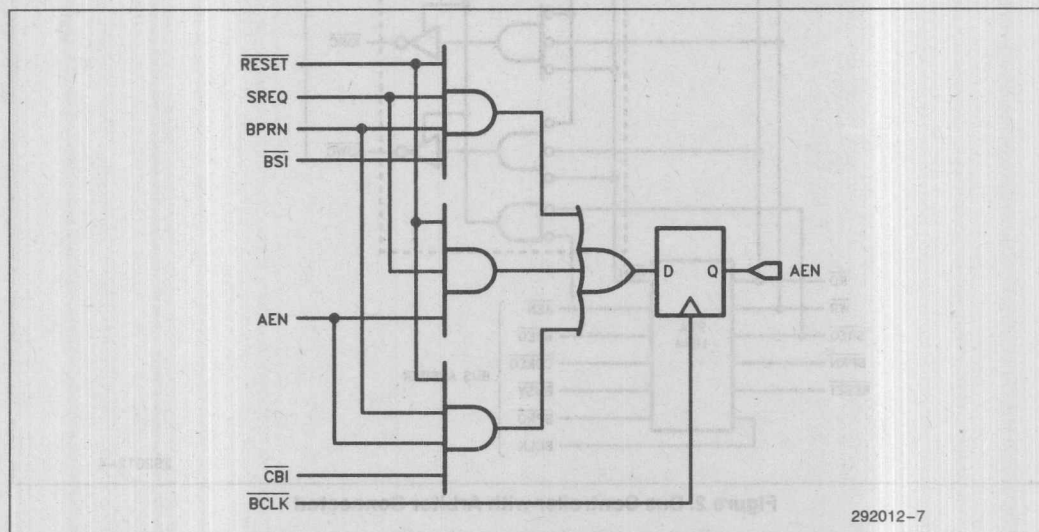
292012-5

Figure 3. List File for PLA Arbiter

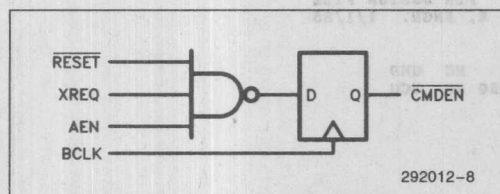




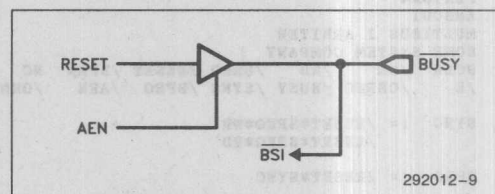
A) Request



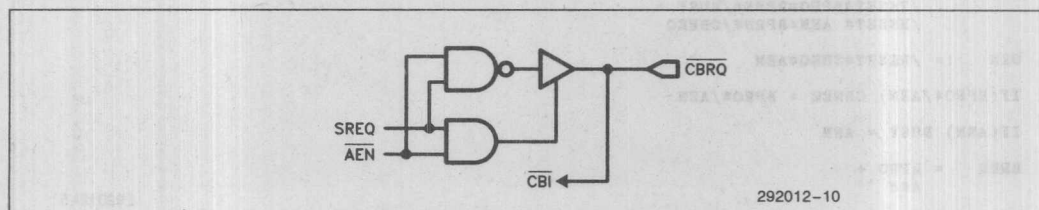
B) Grant



C) Command Enable



D) Busy



E) CBRQ

Figure 4. Logic Diagram of Bus Arbiter Functions

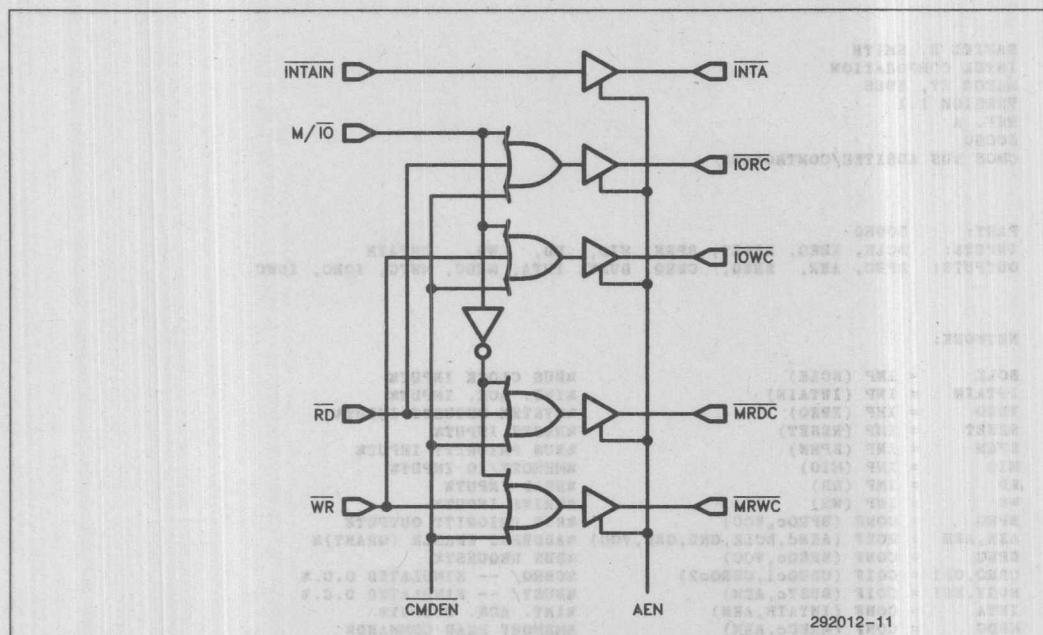


Figure 5. Logic Diagram of Bus Controller Functions

DANIEL E. SMITH  
INTEL CORPORATION  
MARCH 27, 1986  
VERSION 1.1  
REV. A  
5C060

CMOS BUS ARBITER/CONTROLLER

PART: 5C060

INPUTS: BCLK, XREQ, RESET, BPRN, MIO, RD, WR, INTAIN

OUTPUTS: BPRO, AEN, BREQ, CBRQ, BUSY, INTA, MRDC, MWTC, IORC, IOWC

#### NETWORK:

BCLK	= INP (BCLK)	%BUS CLOCK INPUT%
INTAIN	= INP (INTAIN)	%INT. ACK. INPUT%
XREQ	= INP (XREQ)	%SYSTEM REQUEST INPUT%
RESET	= INP (RESET)	%RESET INPUT%
BPRN	= INP (BPRN)	%BUS PRIORITY INPUT%
MIO	= INP (MIO)	%MEMORY/IO INPUT%
RD	= INP (RD)	%READ INPUT%
WR	= INP (WR)	%WRITE INPUT%
BPRO	= CONF (BPROc,VCC)	%BUS PRIORITY OUTPUT%
AEN,AEN	= NORF (AEND,BCLK,GND,GND,VCC)	%ADDRESS ENABLE (GRANT)%
BREQ	= CONF (BREQc,VCC)	%BUS REQUEST%
CBRQ,CBI	= COIF (CBRQc1,CBRQc2)	%CBRQ/ -- SIMULATED O.C.%
BUSY,BSI	= COIF (BUSYc,AEN)	%BUSY/ -- SIMULATED O.C.%
INTA	= CONF (INTAIN,AEN)	%INT. ACK. OUTPUT%
MRDC	= CONF (MRDCc,AEN)	%MEMORY READ COMMAND%
MWTC	= CONF (MWTCc,AEN)	%MEMORY WRITE COMMAND%
IORC	= CONF (IORCc,AEN)	%I/O READ COMMAND%
IOWC	= CONF (IOWCc,AEN)	%I/O WRITE COMMAND%
SREQ	= NORF (SREQd,BCLK,GND,GND)	%VALID BUS REQUEST%
SYNC	= NORF (SYNCd,BCLK,GND,GND)	%SYNCHRONIZED REQUEST%
CMDEN	= NORF (CMDEND,BCLK,GND,GND)	%COMMAND ENABLE%

#### EQUATIONS:

```

BPROc = (SREQ * /BPRN);
AEND = RESET * SREQ * /BPRN * BSI +
      RESET * SREQ * AEN +
      RESET * /BPRN * AEN * CBI;
BREQc = /(SREQ + AEN);
BUSYc = /RESET;
CBRQc1 = /(SREQ * /AEN);
CBRQc2 = SREQ * /AEN;
MRDCc = /MIO + RD + CMDEN;
MWTCc = /MIO + WR + CMDEN;
IORCc = MIO + RD + CMDEN;
IOWCc = MIO + WR + CMDEN;
SREQd = RESET * SYNC;
SYNCd = RESET * XREQ;
CMDEND = /(RESET * XREQ * AEN);

```

END\$

292012-12

292012-13

Figure 6. iPLDS Network List File

Design implemented successfully

DANIEL E. SMITH  
INTEL CORPORATION  
MARCH 27, 1986  
VERSION 1.1  
REV. A  
5C060  
CMOS BUS ARBITER/CONTROLLER

## 5C060

BCLK - 1 24:- Vcc  
MIO - 2 23:- XREQ  
RESERVED - 3 22:- INTA  
RESERVED - 4 21:- IOWC  
RESERVED - 5 20:- IORC  
AEN - 6 19:- MWTC  
BPRO - 7 18:- MRDC  
INTAIN - 8 17:- BUSY  
WR - 9 16:- CBRQ  
RD - 10 15:- BREQ  
BPRN - 11 14:- RESET  
GND - 12 13:- GND

## \*\*INPUTS\*\*

Name	Pin	Resource	MCell #	PTerms	MCells	Feeds:	OE	Clear	Clock
BCLK	1	INP	11	0/ 8	11	1	-	-	CLK1
MIO	2	INP	2	0/ 8	2	3	-	-	-
					4				
					5				
INTAIN	8	INP	11	0/ 8	11	1	-	-	-
WR	9	INP	11	0/ 8	11	2	-	-	-
					4				
RD	10	INP	11	0/ 8	3	5	-	-	-
BPRN	11	INP	-	-	12	13	-	-	-
RESET	14	INP	-	-	6	9	-	-	-
					10	11	-	-	-
					12		-	-	-
XREQ	23	INP	-	-	9	10	-	-	-

292012-14

Figure 7. iPLDS Report File



**OUTPUTS**									
Name	Pin	Resource	MCell #	PTerms	MCells	Feeds:	OE	Clear	Clock
AEN	6	RORF	12	3/ 8	7	-7	-	-	-
					8	1	-	-	-
					9	2	-	-	-
					12	3	-	-	-
						4	-	-	-
						5	-	-	-
						6	-	-	-
BPRO	7	CONF	13	1/ 8	-	-	-	-	-
BREQ	15	CONF	8	1/ 8	-	-	-	-	-
CBRQ	16	COIF	7	1/ 8	12	-	-	-	-
BUSY	17	COIF	6	1/ 8	12	-	-	-	-
MRDC	18	CONF	5	1/ 8	-	-	-	-	-
MWTC	19	CONF	4	1/ 8	-	-	-	-	-
IORC	20	CONF	3	1/ 8	-	-	-	-	-
IOWC	21	CONF	2	1/ 8	-	-	-	-	-
INTA	22	CONF	1	1/ 8	-	-	-	-	-
**BURIED REGISTERS**									
Name	Pin	Resource	MCell #	PTerms	MCells	Feeds:	OE	Clear	Clock
	3	NORF	9	1/ 8	2	-	-	-	-
					3	-	-	-	-
					4	-	-	-	-
					5	-	-	-	-
	4	NORF	10	1/ 8	11	-	-	-	-
	5	NORF	11	1/ 8	7	-	-	-	-
					8	-	-	-	-
					12	-	-	-	-
					13	-	-	-	-
**UNUSED RESOURCES**									
Name	Pin	Resource	MCell	PTerms					
-	13	-	-	-					
**PART UTILIZATION**									
95%	Pins								
100%	MacroCells								
11%	PTerms								

292012-15

Figure 7. iPLDS Report File (Continued)

November 1988

3

# Fitting the 5C180

**TODD KOELLING**  
PROGRAMMABLE LOGIC APPLICATIONS  
INTEL CORPORATION

Order Number: 292053-001

## **FITTING THE 5C180**

## **CONTENTS**

**PAGE**

<b>INTRODUCTION .....</b>	<b>3-191</b>
<b>SUMMIT NUMBER ONE: PIN ESTIMATE .....</b>	<b>3-191</b>
<b>SUMMIT NUMBER TWO: MACROCELL ESTIMATE .....</b>	<b>3-192</b>
<b>SUMMIT NUMBER THREE: SUCCESSFUL TRANSLATION .....</b>	<b>3-193</b>
<b>SUMMIT NUMBER FOUR: REGISTER CLOCK INPUTS .....</b>	<b>3-193</b>
<b>SUMMIT NUMBER FIVE: ASYNCHRONOUS CLOCKS AND OUTPUT ENABLES .....</b>	<b>3-193</b>
<b>SUMMIT NUMBER SIX: GREATER THAN ONE P-TERM REGISTER CONTROLS .....</b>	<b>3-193</b>
<b>SUMMIT NUMBER SEVEN: NOT ENOUGH P-TERMS FOR AN EQUATION .....</b>	<b>3-196</b>
<b>SUMMIT NUMBER EIGHT: MACROCELL RESOURCES EXCEEDED .....</b>	<b>3-197</b>
<b>THE FINAL ASCENT: NOT ENOUGH GLOBAL FEEDBACK .....</b>	<b>3-197</b>
<b>CONCLUSION .....</b>	<b>3-201</b>

## INTRODUCTION

In many ways, fitting the 5C180 is like climbing a mountain. Just when what appears to be the summit is reached, another summit is revealed behind it. This may occur several times before the actual summit is surmounted.

Likewise, fitting a 5C180 may have several false summits. Just when one has conquered what appears to be the "problem", another problem often appears behind it. This may occur several times before the design fitting is complete.

This application note addresses the problems that can be encountered when trying to fit a 5C180 and offers suggestions on how to get past them. The key to the climb is examining what resources are still available after the software\* complains that a particular resource is not available.

## SUMMIT NUMBER ONE: PIN ESTIMATE

Before keying in the design, it is best to estimate the I/O pin requirements. This is done by counting the total number of inputs to the device and outputs from the device.

**PROBLEM:** Not enough Input Pins

**HELP:** Run all synchronous clocks through Clock Buffers (CLKBs). Shared clocks may use the same CLKB output which may result in reduction from 4 CLK input pins to 1 CLK input pin (see Figures 1a &

\*IPLS II ver. 1.1 or later is ESSENTIAL for 5C180 designs as the fitting algorithm was significantly improved with this release.

\*IPLS II ver. 1.5 or later is HIGHLY RECOMMENDED as the error messages and Utilization Report Files were significantly enhanced with this release.

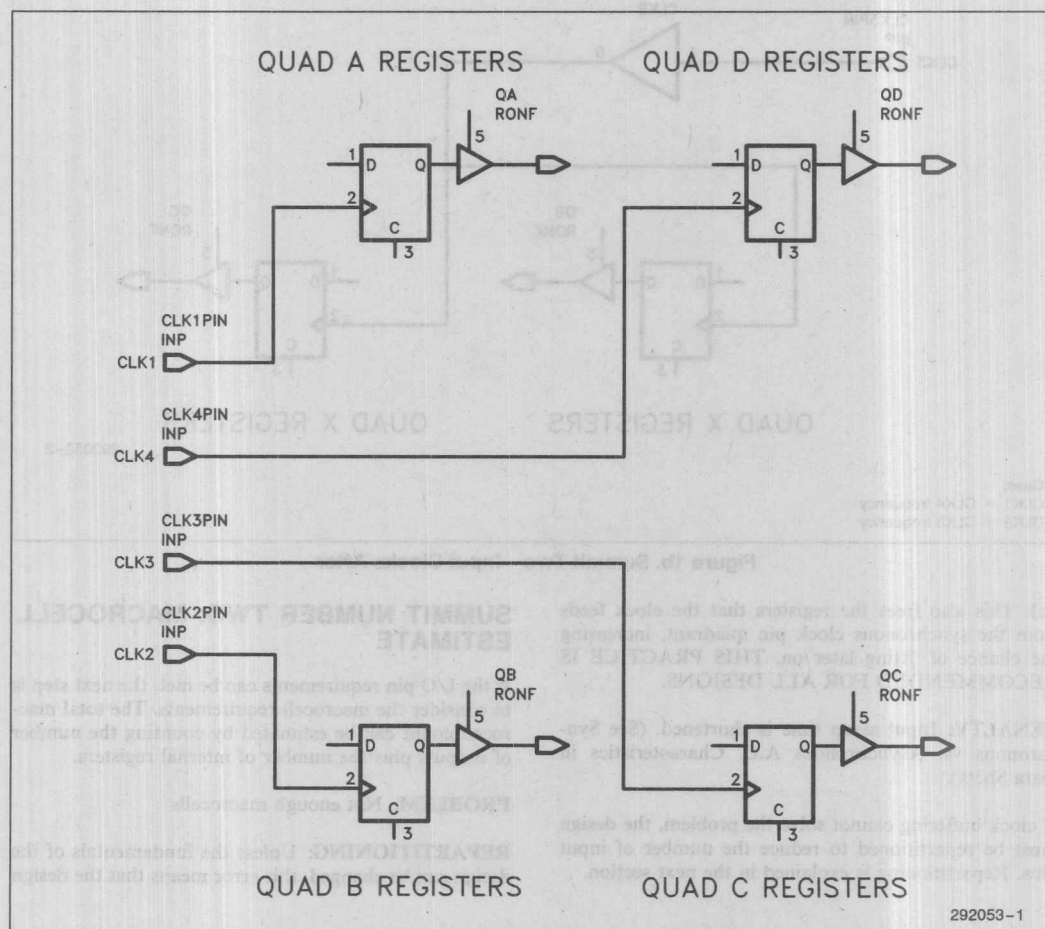


Figure 1a. Summit One—Input Clocks Before



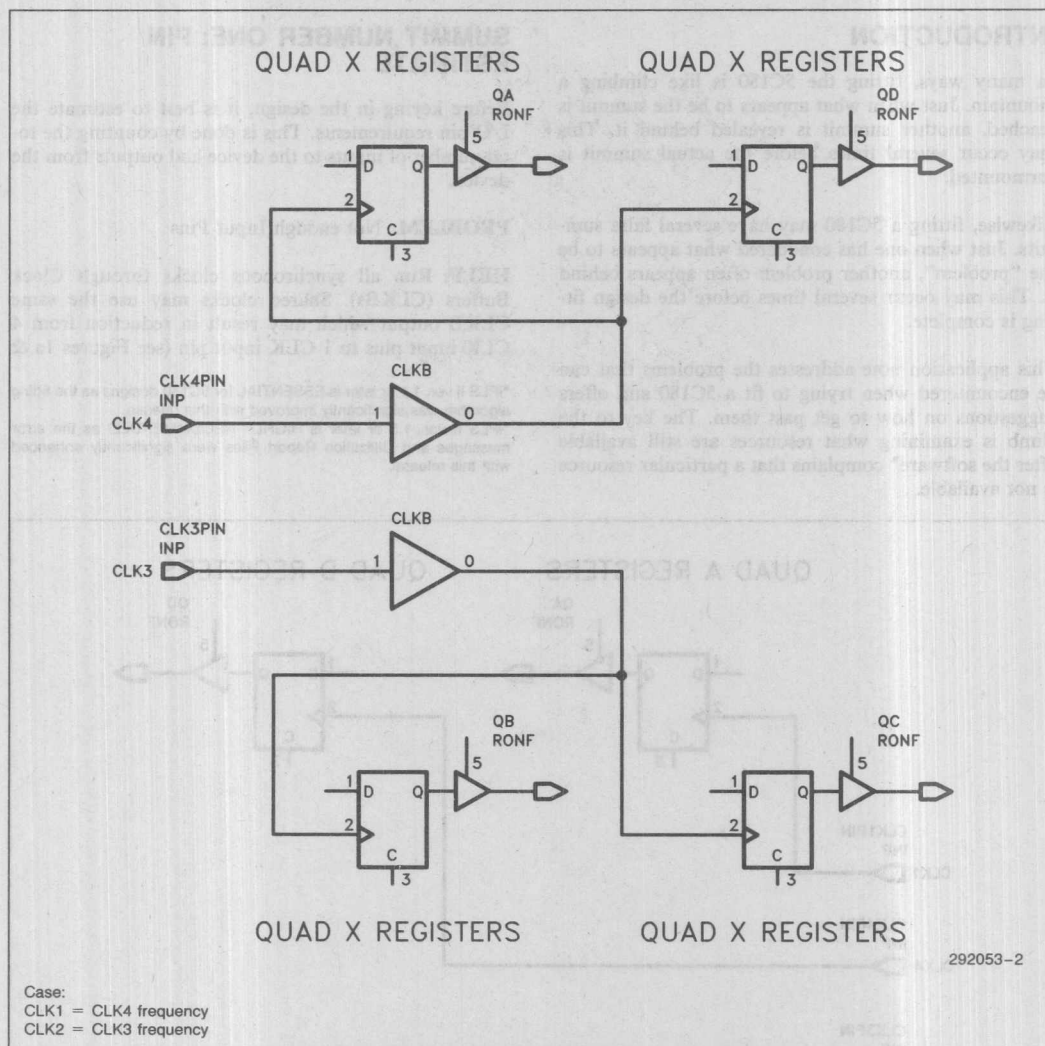


Figure 1b. Summit Two—Input Clocks After

1b). This also frees the registers that the clock feeds from the synchronous clock pin quadrant, increasing the chance of fitting later on. **THIS PRACTICE IS RECOMMENDED FOR ALL DESIGNS.**

**PENALTY:** Input setup time is shortened. (See Synchronous vs. Asynchronous A.C. Characteristics in Data Sheet).

If clock buffering cannot solve the problem, the design must be repartitioned to reduce the number of input pins. Repartitioning is explained in the next section.

## SUMMIT NUMBER TWO: MACROCELL ESTIMATE

If the I/O pin requirements can be met, the next step is to consider the macrocell requirements. The total macrocell count can be estimated by counting the number of outputs plus the number of internal registers.

**PROBLEM:** Not enough macrocells

**REPARTITIONING:** Unless the fundamentals of the design can be changed, this error means that the design

must be repartitioned. This is done by removing part of the circuitry and placing it in a second device such as a 5C060 or 5C090. The 5C060 and 5C090 are recommended since their architectures (and therefore their ADFs) are nearly identical to those of the 5C180 (the NOCF and COCF primitives are the only exceptions).

Portions of the 5C180 ADF can be easily transferred into one of the smaller devices or the smaller device ADFs can be transferred back to the 5C180 if sufficient room is freed up later on. **IT IS RECOMMENDED THAT FOUR OR FIVE UNUSED MACROCELLS BE LEFT IN THE 5C180 FOR USE BY LATER STAGES.**

### SUMMIT NUMBER THREE: SUCCESSFUL TRANSLATION

With the design entered, the next summit is successful translation.

**ERROR:** \*\*\*ERR-MAC-No macrofunction for: ...

**EXPLANATION:** The Macro Expander Module cannot find a macro for a network element.

**FIX:** Make sure correct search path is available for macro libraries. Check for typo or syntax error. If using schematic capture, make certain that only valid EPLD library symbols were entered.

**ERROR:** Any "\*\*\*\*ERROR-XLT-..."

**EXPLANATION:** The Translator found a problem with the way the design was entered. These errors are basically syntax errors which violate ADF format. It may be a simple typo, missing parenthesis or missing semicolon. Remember that the iPLS II LOC does differentiate between upper and lower case letters. If using schematic capture, make sure that all device inputs and outputs have pin symbols and that all the pins and wires are properly labeled.

**FIX:** Refer to your iPLS II manual or call the EPLD Hotline, 1-800-323-EPLD, for help on the tough ones.

### SUMMIT NUMBER FOUR: REGISTER CLOCK INPUTS

**ERROR:** \*\*\*ERROR-XLT-Clock input must be driven by INP or CLKB

**EXPLANATION:** The clock for a flip-flop must be driven synchronously by a direct quadrant clock pin input (INP) or asynchronously through a Clock Buffer (CLKB). This problem occurs when an equation or gate logic is connected directly to the register clock input.

**FIX:** In order to tell the LOC software that the clock for a flip-flop will be driven by an equation or gate logic, a Clock Buffer (CLKB) must be placed between the equation or logic and the register clock input for each register that is asynchronously clocked.

### SUMMIT NUMBER FIVE: ASYNCHRONOUS CLOCKS AND OUTPUT ENABLES

**ERROR:** \*\*\*ERROR-XLT-OE with asynchronous clock not allowed

**EXPLANATION:** Asynchronous clock and output enable can't be used at the same time in the same macrocell. The 5C180 basic macrocell architecture, Figure 2, shows why. A single p-term is shared between the asynchronous clock and the output enable. This means that both switches in the diagram can be up or both switches can be down. By trying to use a p-term output enable with an asynchronous clock, the top switch would have to be down while the bottom switch is up. This cannot be done as then the register would be clocked and enabled with the same signal.

**WORKAROUND:** To get around this problem, one of the signals must be routed through another macrocell (see Figures 3a-b). The clock could be generated in another macrocell, sent out to a pin, then sent back in on the synchronous clock pin. Alternately, in a first macrocell the register is placed as an asynchronously clocked NORF. In a second macrocell, the register feedback is sent out to a pin using a CONF enabled by the desired enable signal.

**PENALTIES:** Routing the clock through a separate macrocell and back in offers slightly better performance—since the synchronous clock to output time is faster than a second macrocell delay, but this implementation uses a lot of resources—three pins and two macrocells. The second method, routing the feedback from the register back and controlling the output enable in a second macrocell is more straightforward and uses less resources.

### SUMMIT NUMBER SIX: GREATER THAN ONE PRODUCT-TERM REGISTER CONTROLS

**ERRORS:** \*\*\*INFO-FIT- Eqn. too big, 4/-1 PTerm(s), on OE signal OE3

\*\*\*INFO-FIT- Illegal inversion of CLEAR input (CLR1)

**EXPLANATION:** As shown in the basic 5C180 macrocell architecture, Figure 2, only one product term (multiple input AND gate) is available for the register

**WORKAROUND:** Once the offending signal has been located, it must be routed through another macrocell using an NOCF primitive (see Figure 4a-b). If the control signal is a clock, then a clock buffer (CLKB) must also be added.

## Clr Fitting Trick

**PROBLEM:** Register clear input breaks 1 p-term resource limit

**TRICK:** If register has D input of either VCC or GND, substitute SR Flip-Flop.

**EXPLANATION:** D-type EPLD register has only 1 AND gate feeding CLR; SR Flip-Flop utilizes logic array for CLR input allowing a max of 8 AND gates (p-terms) for the CLR resource.

**PENALTIES:** SR Flip-Flop is synchronously clocked.  
D register has asynchronous clear.

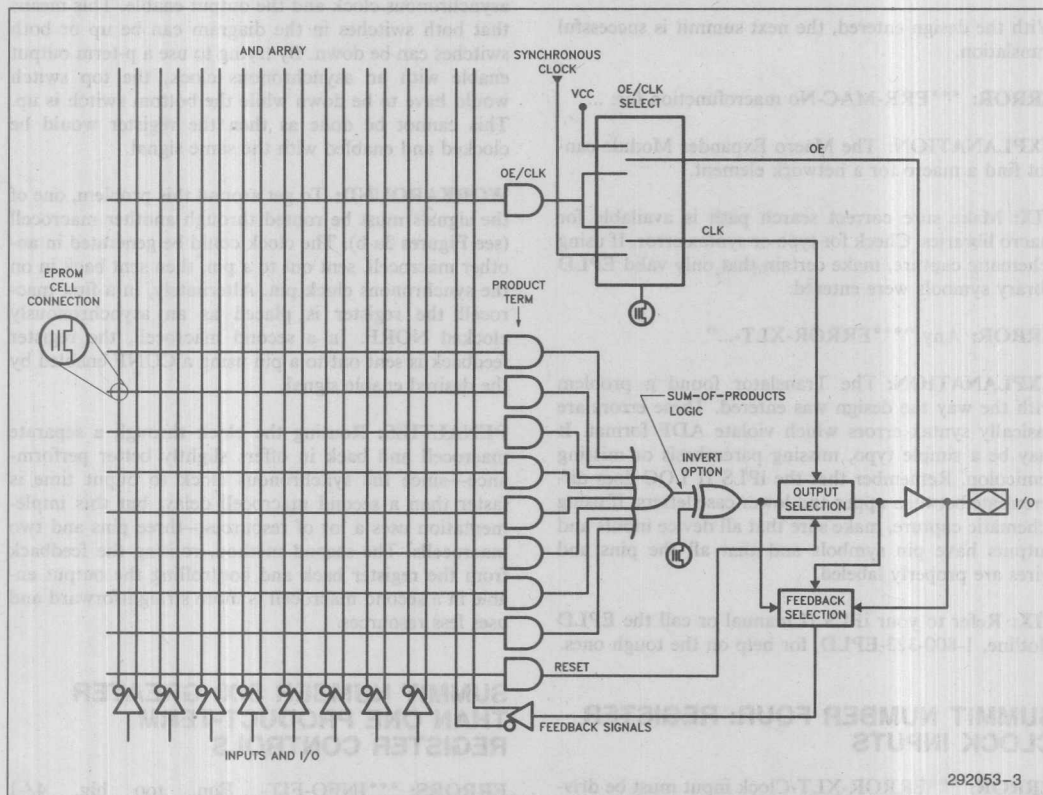
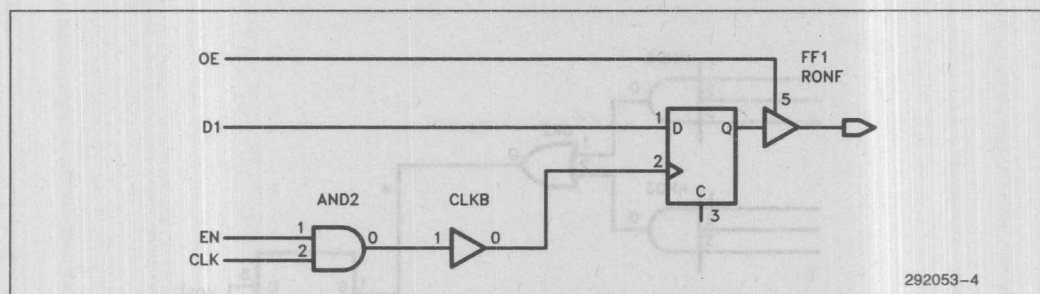
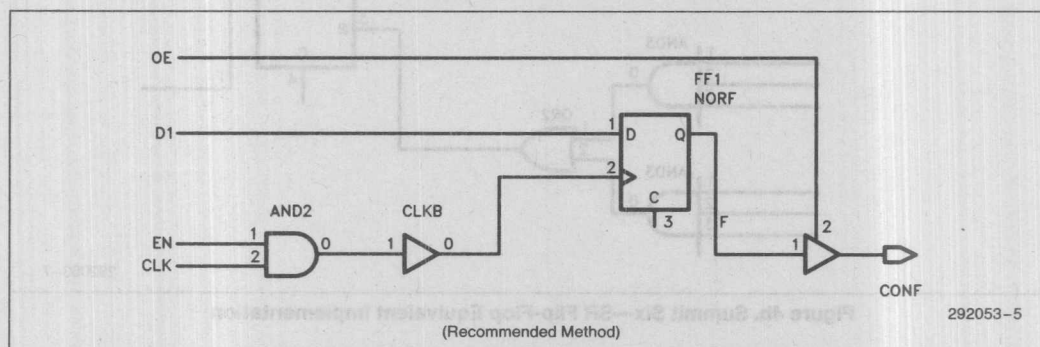


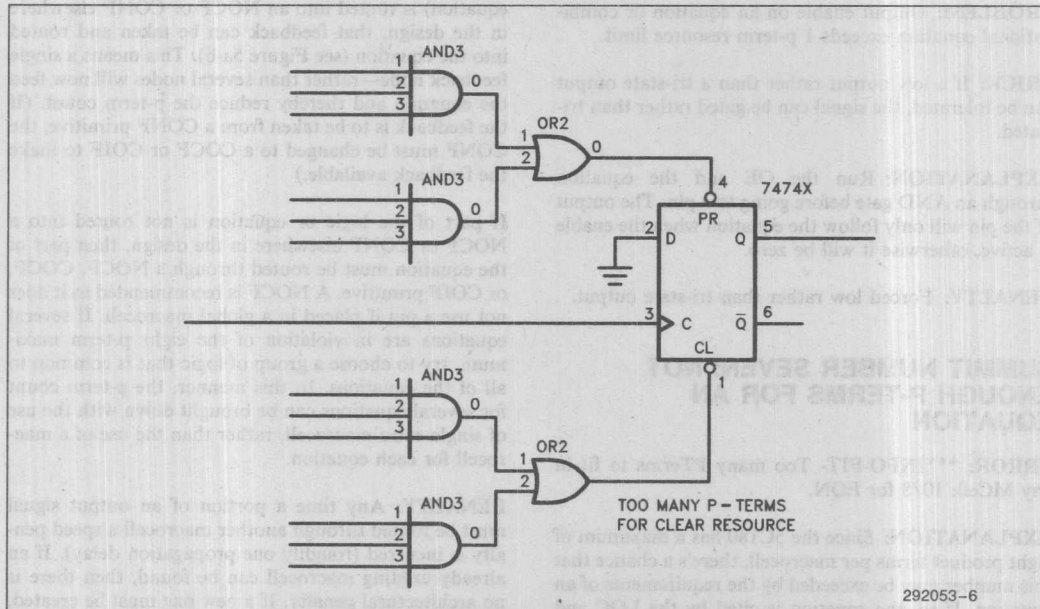
Figure 2. Basic Macrocell Architecture of the 5C180



**Figure 3a. Summit Five—Asynchronous Clock and OE Before**



**Figure 3b. Summit Five—Asynchronous Clock and OE After**



**Figure 4a. Summit Six—Too Many P-Terms on Control - Clear**



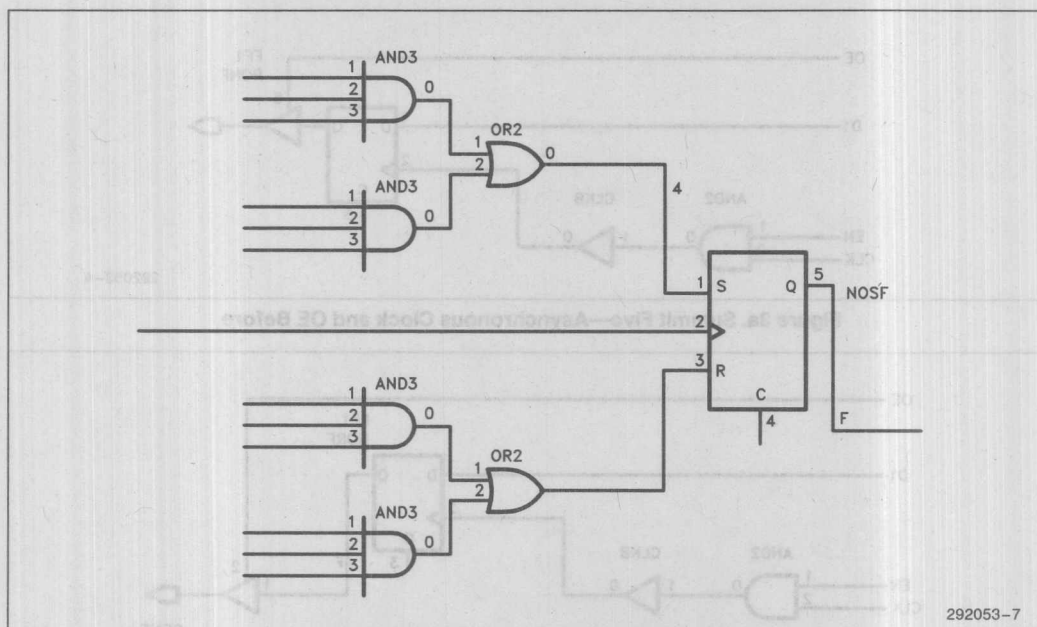


Figure 4b. Summit Six—SR Flip-Flop Equivalent Implementation

### OE Fitting Trick

**PROBLEM:** Output enable on an equation or combinational equation exceeds 1 p-term resource limit.

**TRICK:** If a low output rather than a tri-state output can be tolerated, the signal can be gated rather than tri-stated.

**EXPLANATION:** Run the OE and the equation through an AND gate before going to a pin. The output of the pin will only follow the equation when the enable is active, otherwise it will be zero.

**PENALTY:** Forced low rather than tri-state output.

### SUMMIT NUMBER SEVEN: NOT ENOUGH P-TERMS FOR AN EQUATION

**ERROR:** \*\*\*INFO-FIT- Too many PTerms to fit in any MCell: 10/8 for EQN.

**EXPLANATION:** Since the 5C180 has a maximum of eight product terms per macrocell, there's a chance that this number may be exceeded by the requirements of an equation. If so, the equation is cited by the LOC and can be examined by looking at the EQUATIONS section of the .LEF.

**WORKAROUND:** The workaround for this situation may already be in place! If any portion of the logic (or equation) is routed into an NOCF or CONF elsewhere in the design, that feedback can be taken and routed into the equation (see Figure 5a-b). This means a single feedback node—rather than several nodes will now feed the equation and thereby reduce the p-term count. (If the feedback is to be taken from a CONF primitive, the CONF must be changed to a COCF or COIF to make the feedback available.)

If part of the logic or equation is not routed into a NOCF or CONF elsewhere in the design, then part of the equation must be routed through a NOCF, COCF, or COIF primitive. A NOCF is recommended as it does not use a pin if placed in a global macrocell. If several equations are in violation of the eight p-term maximum, try to choose a group of logic that is common to all of the equations. In this manner, the p-term count for several equations can be brought down with the use of single extra macrocell, rather than the use of a macrocell for each equation.

**PENALTY:** Any time a portion of an output signal must be routed through another macrocell a speed penalty is incurred (roughly one propagation delay). If an already existing macrocell can be found, then there is no architectural penalty. If a new one must be created, then another macrocell is added to the total macrocell count.

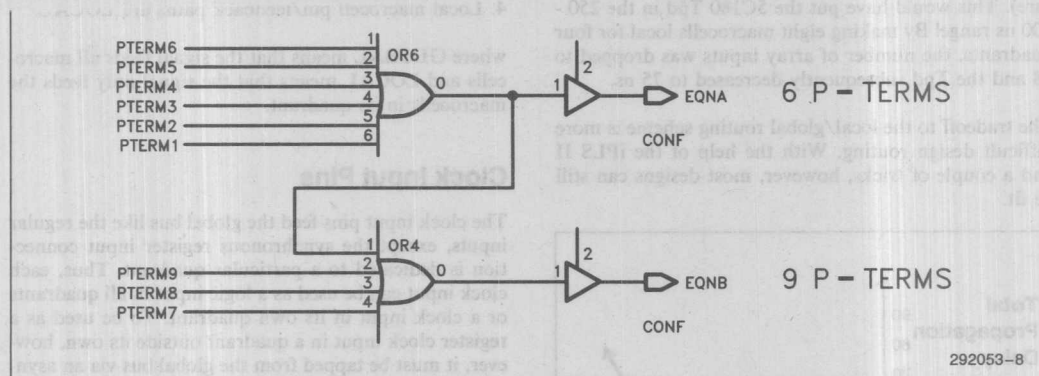


Figure 5a. Summit Seven—Too Many P-Term Equation Before

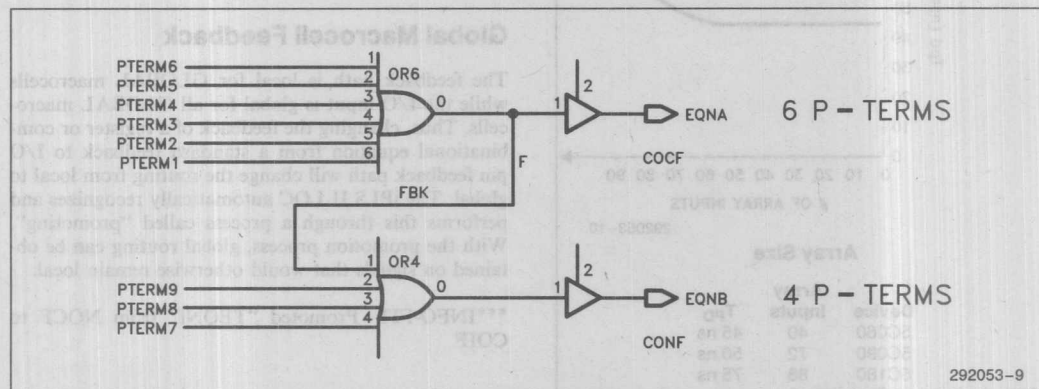


Figure 5b. Summit Seven—Too Many P-Term Equation After

## SUMMIT EIGHT: MACROCELL RESOURCES EXCEEDED

**ERROR: \*\*\*INFO-FIT-** Design requires too many macrocells

**EXPLANATION:** If this error didn't occur at the beginning, there's a good chance summits five, six or seven will push the macrocell count over the limit. (Remember that the macrocell count includes not only the outputs, but also the buried resources such as NOCFs, NORFs and NOTFs). To find out exactly how many macrocells the design requires, LOOK AT THE NETWORK: SECTION OF THE LOGIC EQUATION FILE (.LEF). The inputs list in the LEF will list both the outputs and all the buried resources required by the design. If the count exceeds 48, then too many macrocells are required.

**FIX:** Repartition. The same applies if the number of input pins is exceeded.

## THE FINAL ASCENT: NOT ENOUGH GLOBAL FEEDBACK!

Congratulations! If you have made it this far, you have demonstrated courage, intelligence and tenacity beyond that of the average climber. You will soon be rewarded, but first there is one more obstacle to be overcome. Welcome to the North Face of local/global feedback!

## A Word About Local/Global Feedback

First of all, why does local/global feedback exist? The answer can be found in the graph shown in Figure 6. The propagation delay versus array size is shown for the 5C060/090/180 family. As the number of inputs into the array increases, the propagation delay increases...exponentially. If all the inputs and feedback were made global, the 5C180 would have 136 inputs feeding each array (remember that both true and complement polarities must be fed into the array of a PLD architec-

ture). This would have put the 5C180 Tpd in the 250 - 300 ns range! By making eight macrocells local for four quadrants, the number of array inputs was dropped to 88 and the Tpd subsequently decreased to 75 ns.

The tradeoff to the local/global routing scheme is more difficult design routing. With the help of the iPLS II and a couple of tricks, however, most designs can still be fit.

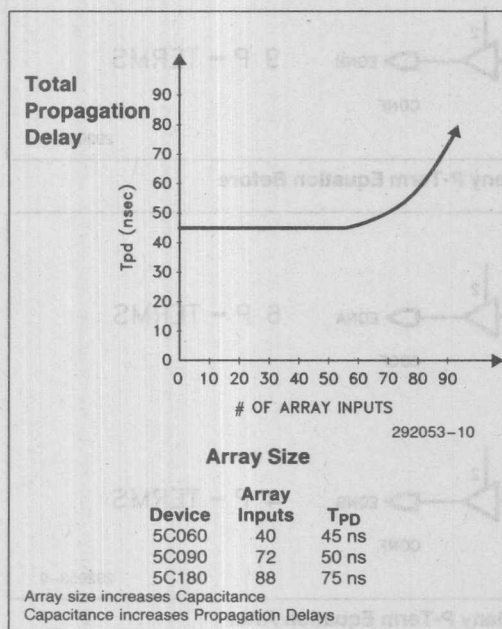


Figure 6. Propagation Delay vs. Array Size for the 5C060/090/180 Family

## A Few Notes

The global/local macrocell assignments are shown in Figure 7. Please note that:

1. Dedicated input pins are GLOBAL.
2. Global macrocell I/O pin are GLOBAL.
3. Global macrocell internal feedback paths are LOCAL.

4. Local macrocell pin/feedback paths are LOCAL.

where GLOBAL means that the signal feeds all macrocells and LOCAL means that the signal only feeds the macrocells in its quadrant.

## Clock Input Pins

The clock input pins feed the global bus like the regular inputs, except the synchronous register input connection is dedicated to a particular quadrant. Thus, each clock input can be used as a logic input in all quadrants or a clock input in its own quadrant. To be used as a register clock input in a quadrant outside its own, however, it must be tapped from the global bus via an asynchronous clock buffer (CLKB).

## Global Macrocell Feedback

The feedback path is local for GLOBAL macrocells while the I/O input is global for all GLOBAL macrocells. Thus, changing the feedback of a register or combinational equation from a standard feedback to I/O pin feedback path will change the routing from local to global. The iPLS II LOC automatically recognizes and performs this through a process called "promoting". With the promotion process, global routing can be obtained on signals that would otherwise remain local.

\*\*\*INFO-FIT- Promoted "TEQNF" from NOCF to COIF

## Burying a Register in a Global Cell

Because the global macrocells have separate register and I/O pin feedback paths, it is possible to "bury" a register or equation by disabling the output buffer and still use the pin as an input. The iPLS II LOC automatically assigns an input to the pin of a buried register macrocell if it is necessary and possible. Such assignments are documented in the Utilization Report File (.RPT). If manual assignment is desired, it may be performed by placing the input pin assignment in the ADF INPUTS: list and assigning the buried register feedback to the same pin in the OUTPUTS: list (Figure 8). Registers or equations can only be buried on global macrocells, since local macrocells only have one feedback path that is used for either the register or the pin feedback.

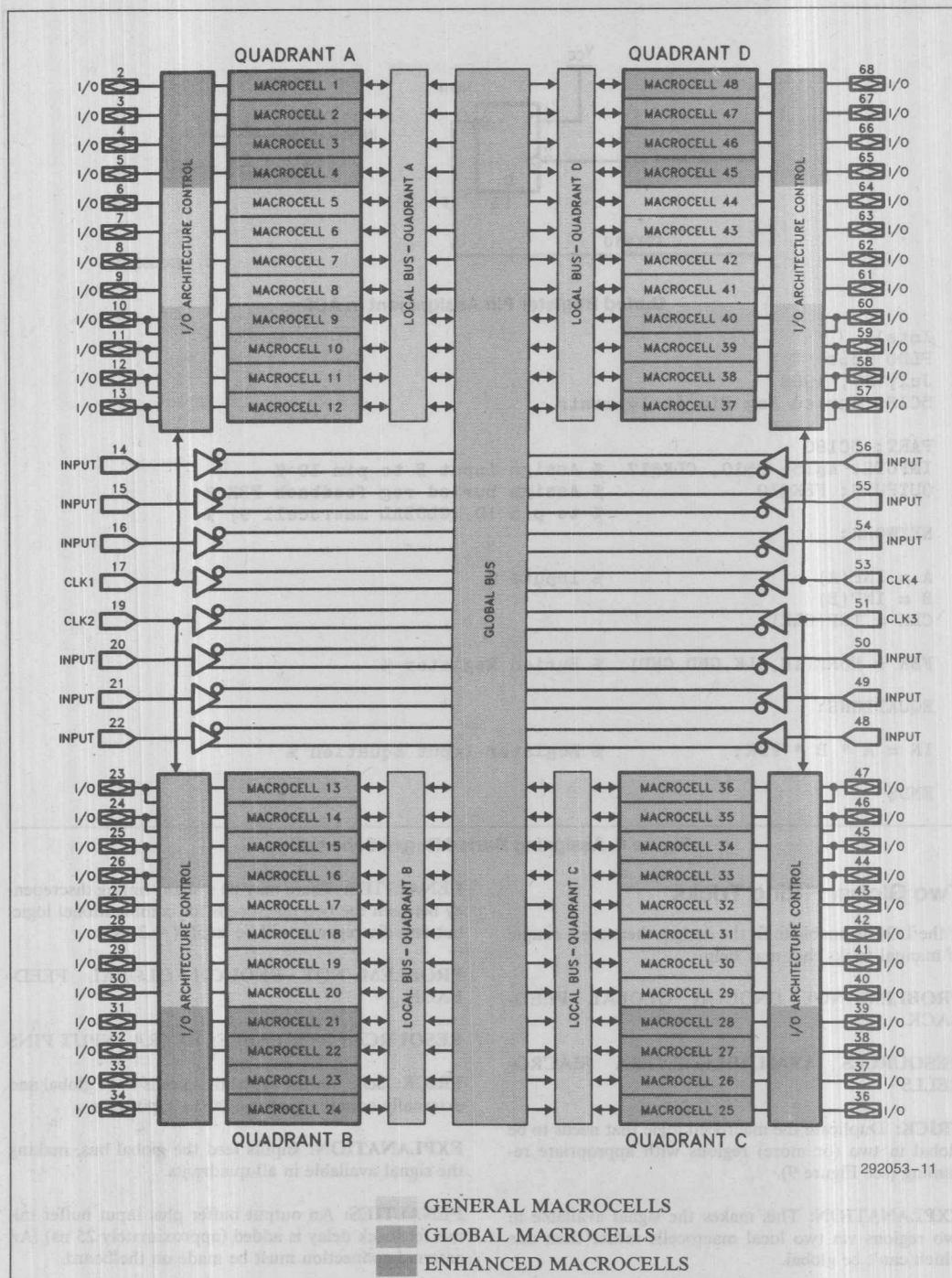
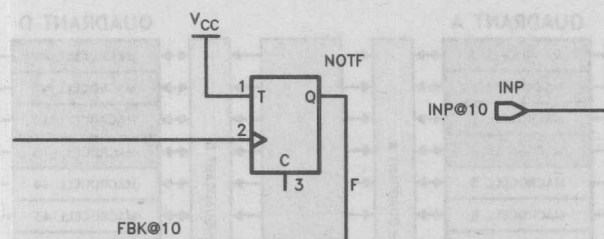


Figure 7. 5C180 Block Diagram





292053-12

### Buried Register Pin Assignment in ADF

Intel  
PLDO Apps  
July 27, 1988  
5C180 Buried Reg Pin Assignments

PART: 5C180  
INPUTS: A@15, B@10, CLK@17 % Assign input B to pin 10 %  
OUTPUTS: FBK@10 % Assign buried reg feedback FBK %  
% to pin 10 (GLOBAL macrocell 9) %

#### NETWORK:

A = INP(A) % Inputs %  
B = INP(B)  
CLK = INP(CLK)

FBK = NORF(IN,CLK,GND,GND) % Buried Register %

#### EQUATIONS:

IN = A \* B \* FBK; % Register Input Equation %

END\$

Figure 8. Assigning Buried Reg in Schematic

## Two Global Fitting Tricks

If the LOC is unable to fit the design, there are a couple of manual tricks that may help:

### PROBLEM: NOT ENOUGH GLOBAL FEEDBACK

### RESOURCES AVAILABLE: EXTRA MACROCELLS

**TRICK:** Duplicate the macrocell logic that needs to be global in two (or more) regions with appropriate renaming (see Figure 9).

**EXPLANATION:** This makes the signal available in two regions via two local macrocells rather than one which can't be global.

**PENALTIES:** There may be a slight timing discrepancy between the two macrocells for combinational logic, but any discrepancy will be small (< 2 ns).

### PROBLEM: NOT ENOUGH GLOBAL FEEDBACK

### RESOURCES AVAILABLE: EXTRA INPUT PINS

**TRICK:** Send out the signal that needs to be global and externally connect it to one of the input pins.

**EXPLANATION:** Inputs feed the global bus, making the signal available in all quadrants.

**PENALTIES:** An output buffer plus input buffer minus feedback delay is added (approximately 25 ns). An external connection must be made on the board.

# NOTE:

For the previous tricks, look at the Utilization Report (.RPT) file. The "Interconnect Cross Reference" is particularly useful for examining the routing requirements of the design.

If the previous tricks cannot be done (see Figure 11) and scrutinization of the Interconnect Cross Reference reveals no other way to achieve the desired routing, repartitioning is necessary. That is, place a chunk of interconnected logic into a 5C060 or 5C090 and go back to the start.

# CONCLUSION

Fitting the 5C180 is a process with many stages. One difficulty may hide the next and fixing one problem will sometimes uncover another. Equipped with the iPLS II LOC and a few tricks, however, fitting can be accomplished.

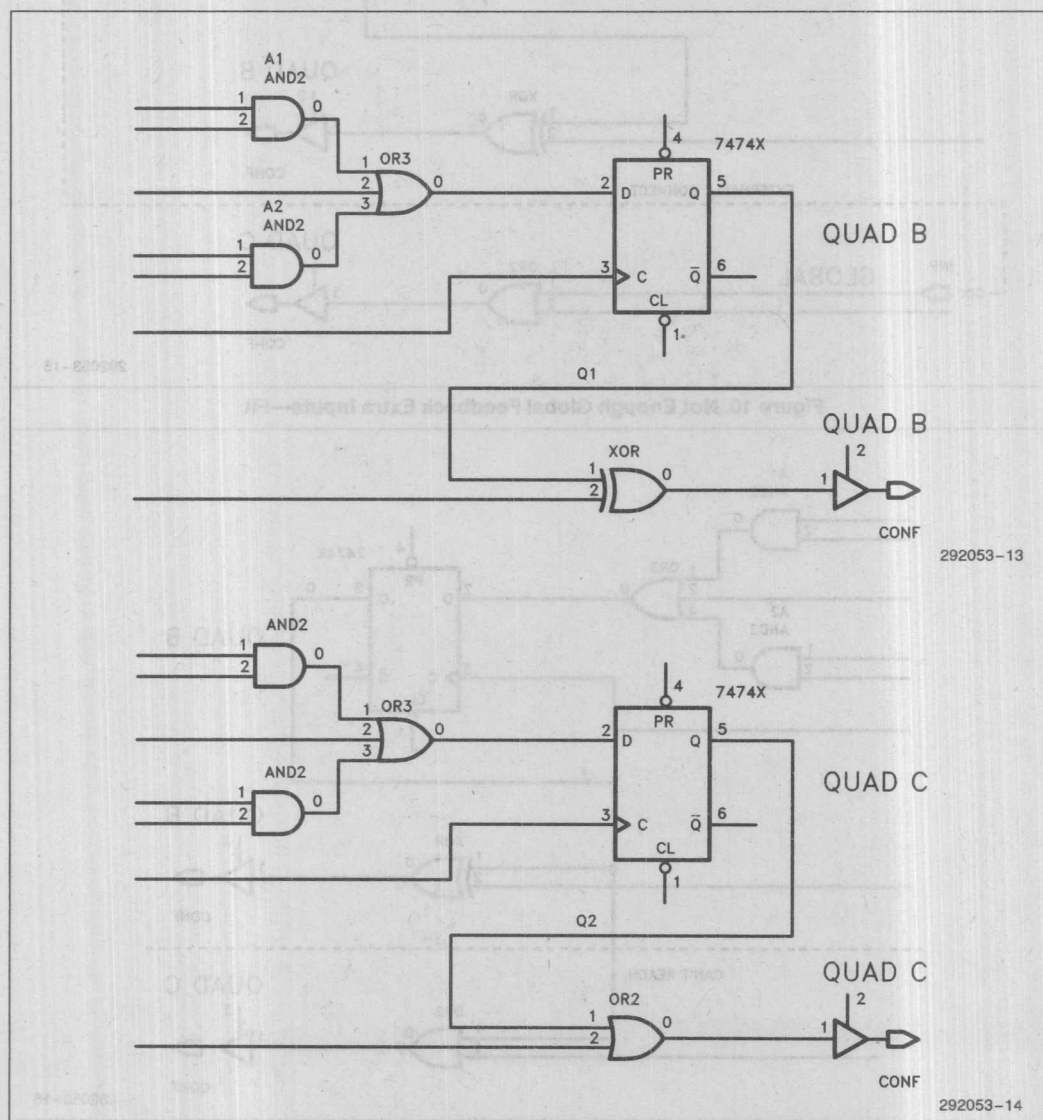


Figure 9. Not Enough Global Feedback Extra Macrocells—Fit



**Figure 10. Not Enough Global Feedback Extra Inputs—Fit**



### Figure 11. Not Enough Global Feedback—No Fit

January 1987

3

# **EPLDs, PLAs and TTL Comparing the "Hidden Costs" in Production**

**PEDRO VARGAS**  
PROGRAMMABLE LOGIC APPLICATIONS  
INTEL CORPORATION

Order Number: 292030-001



# **EPLDs, PLAs AND TTL COMPARING THE "HIDDEN COSTS" IN PRODUCTION**

<b>CONTENTS</b>	<b>PAGE</b>
<b>INTRODUCTION</b> .....	3-205
<b>OBJECTIVE</b> .....	3-205
<b>COSTS DEFINED</b> .....	3-205
<b>ARBITER CIRCUIT</b> .....	3-207
Implementation Requirements .....	3-207
Production Costs .....	3-208
Components .....	3-208
Incoming Inspection .....	3-209
Inventory .....	3-210
PCB Fabrication .....	3-210
Traces .....	3-211
Assembly .....	3-211
Test .....	3-211
Rework .....	3-212
Quality Control .....	3-213
Power Supply .....	3-213
Additional Costs .....	3-213
Programming Loss .....	3-213
Programming Fee .....	3-214
Safety Stock .....	3-214
De-Coupling Capacitors .....	3-214
Other Costs To Consider .....	3-215
Defect Escapes .....	3-215
Cables/Wiring Harness .....	3-215
Enclosure .....	3-215
Arbiter Circuit Conclusion .....	3-216
Development Costs .....	3-218
Research .....	3-218
Prototyping .....	3-218
Debugging .....	3-218
PCB Layout .....	3-220
<b>WINDOW CIRCUIT</b> .....	3-220
Background Information .....	3-220
Production Costs .....	3-220
Window Circuit Conclusion .....	3-224
<b>SUMMARY</b> .....	3-224
<b>REFERENCES</b> .....	3-224

## INTRODUCTION

When comparing logic alternatives, too often the outcome is dominated by the piece price of the components. A side by side comparison based on component costs only, may give the appearance that EPLDs are cost prohibitive. However, when the overall cost of manufacturing a system is considered, the higher integration of EPLDs proves to be a cost-effective solution.

## OBJECTIVE

This application note examines the total costs associated with designing, prototyping, and manufacturing a system. Once these costs have been examined, a comparison is made between EPLDs and other logic alternatives. By being aware of these additional costs, the engineer can make a more accurate cost comparison as a design is begun.

## COSTS DEFINED

Costs can be difficult to pinpoint, let alone measure. However, with a bit of examination, we can break down costs into the following categories;

- Design costs — the cost of conceiving a product

- Prototype costs — first implementation of the product idea
- Production costs — volume manufacturing of the product

Usually, the brunt of the cost for the first two categories is dismissed as NRE (non recurring expense). The effect of these costs on the overall project is examined later, let's look at the third category. Production costs, can be further broken down into;

- Component costs — the cost of the parts per board
- Inspection costs — labor costs for receiving the parts
- Inventory costs — the cost for storing, handling and dispensing the parts
- PCB fabrication — the cost for labor and equipment used in building a board
- Integration costs — the cost of harnesses, enclosures, nuts and bolts etc.

It's important to understand how the cost of a product is affected not only by the cost of the ICs used, but also by the other costs listed above. Figure 1 is a graph which shows this relationship.

3

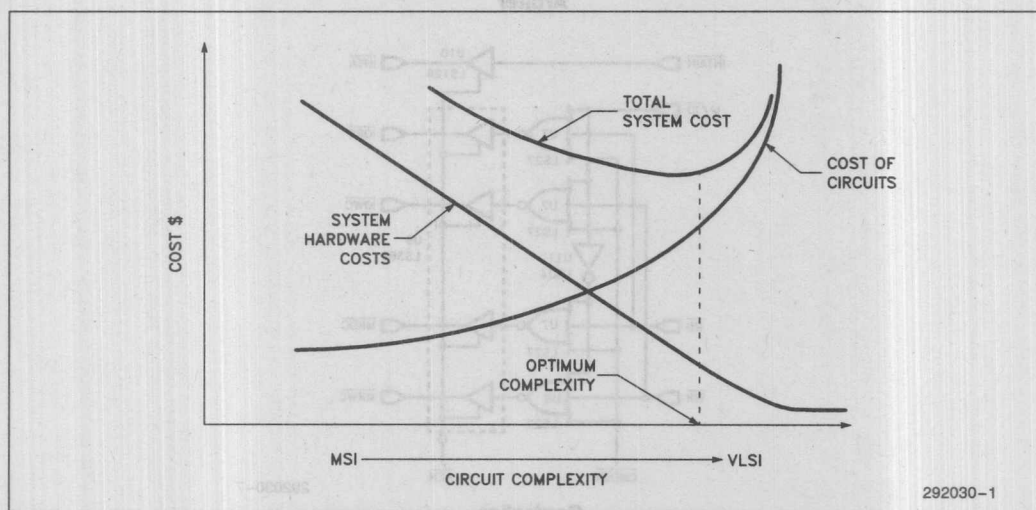


Figure 1. Optimizing Circuit Complexity

292030-1







Table 1. Arbiter/Controller TTL Component List

IC	Type	DIP	I <sub>CC</sub> (mA)	Area (in <sup>2</sup> )	Cost \$
U1	LS08	14 PIN	8.8	0.21	0.18
U2	LS74	14 PIN	8	0.21	0.24
U3	LS21	14 PIN	4.4	0.21	0.22
U4	LS10	14 PIN	3.3	0.21	0.16
U5	LS11	14 PIN	6.6	0.21	0.22
U6	LS02	14 PIN	5.4	0.21	0.17
U7	LS27	14 PIN	6.8	0.21	0.23
U8	LS27	14 PIN	6.8	0.21	0.23
U9	LS366	16 PIN	21	0.24	0.39
U10	LS126	14 PIN	22	0.21	0.39
U11	LS04	14 PIN	6.6	0.21	0.16

The PAL version of the circuit is shown in Figure 3. Two PALs are used due to the requirement of registered outputs on several of the signals.<sup>[20]</sup>

The complete circuit can also be designed in one 5C060 EPLD (Figure 4).<sup>[18]</sup> Looking at the three figures quickly points out the amount of circuit board space required by each version. The three implementations are compared side by side in Table 2.

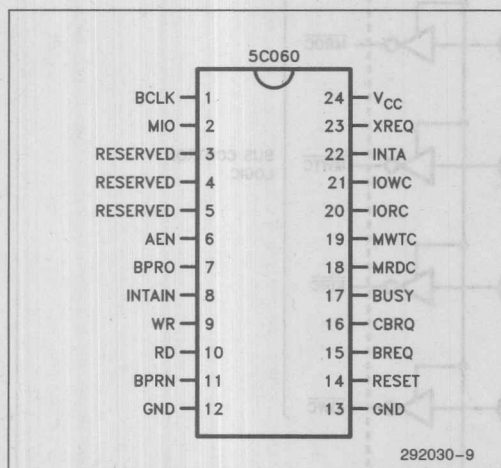


Figure 4. MULTIBUS Arbiter/Controller-EPLD Implementation

Table 2. Implementation Results for Arbiter/Controller

Item	TTL	PLA	EPLD
IC Count	11	2	1
Pin Count	156	40	24
Interconn	36	7	0
Area	2.34	0.6	0.36
I <sub>CC</sub> (mA)	100	240	15
P <sub>wr</sub> (mW)	500	1,200	75

- IC Count — The total chip count
- Pin Count — The total number of IC pins
- Interconnections — The traces required to connect logic gates together
- Area (inches-square) — The sum of the area of all ICs
- I<sub>CC</sub> (mA) — The current consumed while active
- P<sub>wr</sub> (mW) — Total power consumption at 5 VDC.

## Production Costs

Earlier, we noted that production costs consist of many variables. Usually, these variables are lumped together under the term "hidden cost". Although hidden costs are kept in mind by engineers, lack of tangible figures usually precludes their use in detailed cost breakdowns. For this reason, several manufacturers and consulting firms have come up with typical costs per IC and per pin.

For example, SOURCE III (San Jose, CA) reports in one of their studies that the manufacturing cost of a system translates to about 0.35 cents per IC pin. ICE Corporation (Scottsdale, AZ) and EDN magazine concur that the inserted cost of an IC is about \$2 dollars. DATAQUEST also published a cost of about \$2 to \$4 per IC. While the data seems to be consistent, most engineers want to see for themselves how figures like these might be arrived at. The next sections provide insight into this process.

## COMPONENTS

The cost of the component is the easiest value to obtain. A quick call to a distributor or (at worst) a scan through the back of BYTE magazine (for TTL) gives us this cost. Table 3 shows the breakdown of component costs for each version of our MULTIBUS I circuit.

Table 3. Average Component Costs

Package	TTL	PLA	EPLD
DIP14	\$0.25		
DIP16	\$0.35		
DIP20	\$0.55	\$1.50	
DIP24		\$2.90	\$6.00

The price of TTL has changed very little for the last few years<sup>[24]</sup> while EPLDs are dropping in price tremendously. PALs have also leveled off in pricing. Why? Figure 5 shows the life cycle curve of IC products used by the semiconductor industry. From the curve we see that TTL is in the stable range and prices are not likely to drop much more. PALs are also maturing and approaching a stable pricing range. EPLDs however, are in a growth area and historically this is

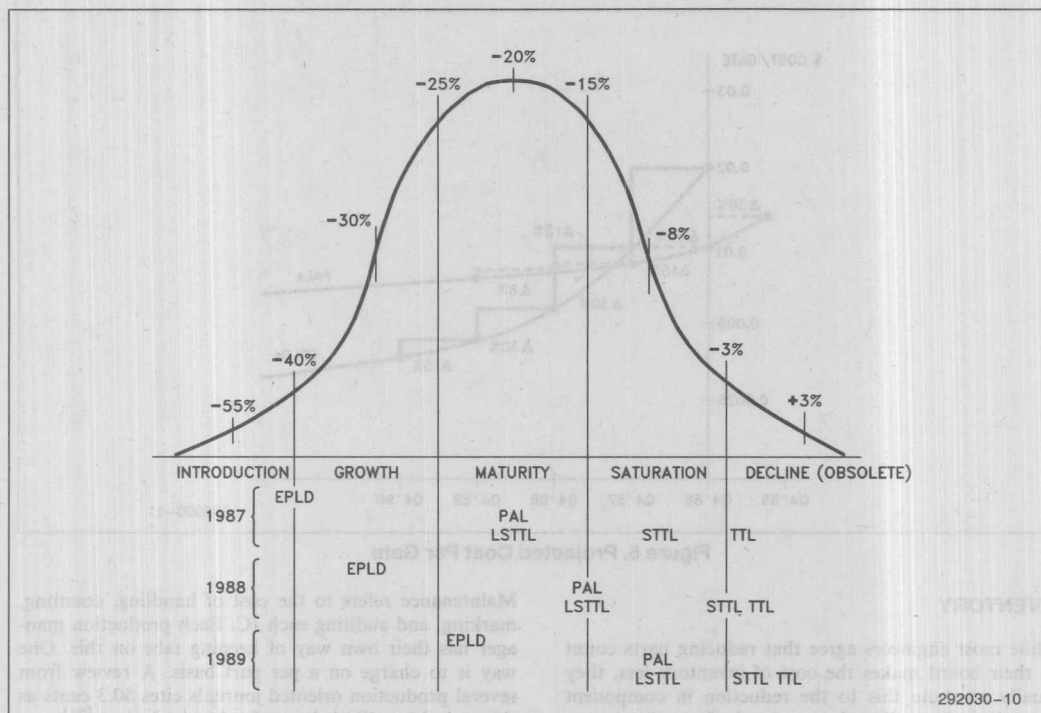


Figure 5. Typical Price Changes Through Semiconductor Product Life Cycle

where the heaviest pricing pressure is. This means that while EPLDs might be expensive (per part) right now, it's not out of the question to expect a 30% per year price reduction as the process is honed and perfected. In other words, it's also important to consider the price of a component at the projected production date, not just at design time.

Life cycle position is also important in understanding the gate cost that is associated with programmable logic devices like PALs and EPLDs. This relationship is shown in Figure 6. The curves translate our observation that newer devices have steeper price cuts during their introduction phase. The PAL curve shows that the cost per gate is leveling off due to the maturity of the device. In contrast, the EPLD is in the growth region, and based on the traditional price reductions, shows a cost per gate that intersects and bypasses the PAL curve.

### INCOMING INSPECTION

For most companies, incoming inspection is more than taking the parts and putting them on the shelf. Most have visual checking as well as some form of IC testing. The variables here are, what amount of human intervention is needed, are automatic handlers needed, are "go/no go" tests or "binning" done automatically? The typical scenario means that components are graded and tested individually, and then placed into one of several bins or kitted. Because the operators handle a large variety of pinned devices (resistors, capacitors, ICs), the cost can be distributed on a per pin basis. Many companies use a penny per pin for this cost.<sup>[16]</sup>

$$\text{Inspection cost} = \$0.01 \text{ per pin}$$

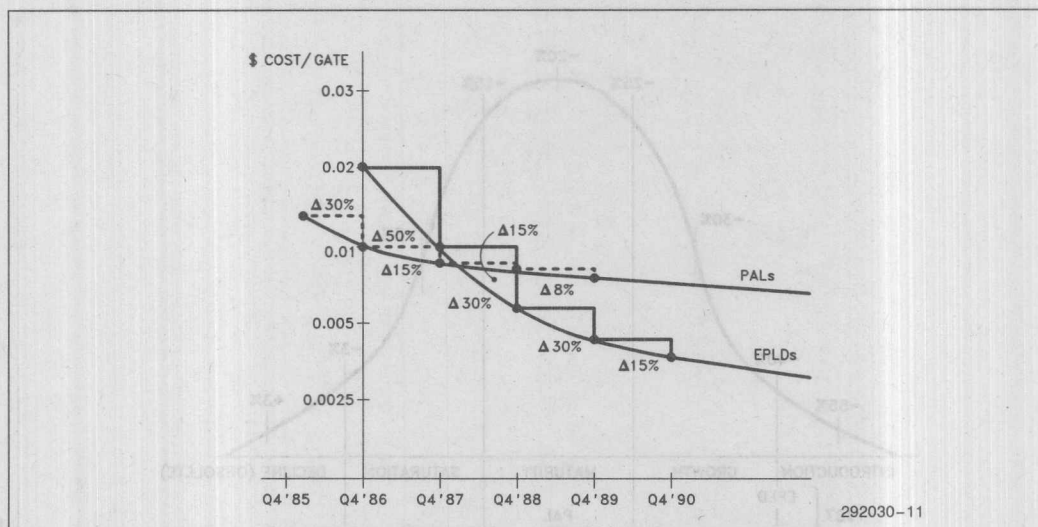


Figure 6. Projected Cost Per Gate

## INVENTORY

While most engineers agree that reducing parts count on their board makes the cost of inventory less, they usually attribute this to the reduction in component costs alone. In reality, the overhead of carrying inventory is made up of the following factors:[21]

- Cost of the component
- Cost of storage
- Maintenance costs
- Data processing
- Usage
- Taxes insurance and interest
- Turnover rate

The American Production and Inventory Control Society (APICS) reports that since 1973 the median cost of carrying inventory has been about 25% of total production costs. They also note that the largest contributing factors are the cost of materials handling storage, and data processing. For simplicity, let's limit our inventory cost to these items.

$$\text{Inventory cost} = \text{storage} + \text{maintenance} + \text{processing}$$

Depending on the locale of a company, the cost of storage can vary greatly. However, this cost is charged on a square foot per year basis. Lets assume a conservative figure of \$20 dollars and distribute this among the ICs in our example circuit.

$$\text{storage} = [\text{Total IC area (sq. ft.)} \times \$20] / \text{IC count}$$

Maintenance refers to the cost of handling, counting, marking, and auditing each IC. Each production manager has their own way of keeping tabs on this. One way is to charge on a per part basis. A review from several production oriented journals cites \$0.3 cents as the typical handling charge for 16 pin devices.[23]

$$\text{Maintenance} = \$0.03 \text{ per 16 pin part.}$$

Processing[21] usually entails a parts log that tracks each part by manufacturer, cost, second source etc. Also, monthly shortage reports are quite common as are quarterly orders and audits. Limiting this cost to paper only, at one sheet of paper per week, per year, at a cost of a penny per part type;

$$\text{Processing} = \$0.52 \text{ per part type per year}$$

## PCB FABRICATION

The cost of manufacturing (cutting, etching, drilling) a circuit board seems to vary around two pricing methods. Some fab houses charge on a square inch basis. Others base their price on a gut feeling based on previous jobs. The square inch method is the most common.

Items of interest in evaluating PCB costs are, number of ICs, number of traces and vias, and in general, the complexity of the board. Traces that are smaller than 10 mils require extra care in etching. Depending on complexity, and additional charge might be added to the area cost. This charge covers material loss in case of low etch yields. Yield is directly dependent on the number of ICs on a board. In other words, more ICs mean more holes, tighter traces, and a greater chance of losing some boards in their processing. The average going

The price increases by about 40% for every two layers. This extra charge, however is too subjective to consider in our comparison.

$$\text{PCB Fab} = [\$0.20 \times \text{total IC area (sq. inch)}] / \text{IC count}$$

## Traces

There is a real cost involved with traces, which doesn't surface until later in the production cycle or on a later board revision. A technical paper presented at the 1984 international Test Conference<sup>[1]</sup> estimates that the cost of a trace on a board is ten to thirty times that of one made in silicon. The cost of traces is taken up by:

- Increased drilling (more traces = more vias = more holes)
- Lower PCB yield (smaller mill lines drop the board yield)
- Increased risk of trace to trace shorts (lower reliability)
- More expensive artwork mods (it costs more to move traces around on a board)
- More expensive PCB mods (cost of cuts, jumpers, and rework)

In our circuit example, an extra trace is that which is unnecessary in contrasting implementations. For example, referring to Figure 2, of all the traces required to connect/RESET in the TTL implementation, only one will be required for the EPLD and PAL circuit (the input); the others won't be needed.

For our comparison, let's take the median value of twenty as our multiplying factor. Since a silicon trace costs an order of magnitude less than an EPLD gate (\$0.01), the resulting cost of a PCB trace is;

$$(\$0.01/10) \times 20 = \$0.02 \text{ cents per trace}$$

$$\text{Trace cost} = [\text{total trace count} \times \$0.02] / \text{IC count}$$

The cost of assembling a board is largely dependent on labor charges and capital. Assembly consists of lead forming, component insertion, and soldering. The labor charge is hourly and varies between domestic and off-shore assembly houses. While machines can certainly do lead cutting, crimping, and insertion, human intervention is still an expensive presence. Assembly costs can be charged on a per board or per chip basis. The latter is more appropriate for our comparison. The average charge (domestically) is about \$0.10 per IC.

$$\text{Assembly} = \$0.10 \text{ per 16 pin part}$$

One important result of using high integration parts like EPLDs is that the assembly procedures (manual or automatic) go smoother. This is due to fewer parts being handled, and less overheating of the equipment. Overall, the industry reports less insertion faults (parts stuffed wrong) as denser ICs are used and as insertion equipment matures with them.

## TEST

Test strategies can vary, but the typical test flow for a board<sup>[3]</sup> is shown in Figure 7. The process is basically taking a board through increasing complexity levels of testing. For example, ATE might be a bed of nails fixture that catches 60 percent of the faults. Test bed is usually a backplane with all boards known good except for the one under test. System test is the final integration of all the boards that were tested individually.

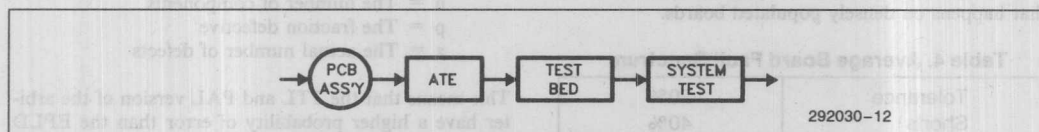


Figure 7. Typical Test Flow



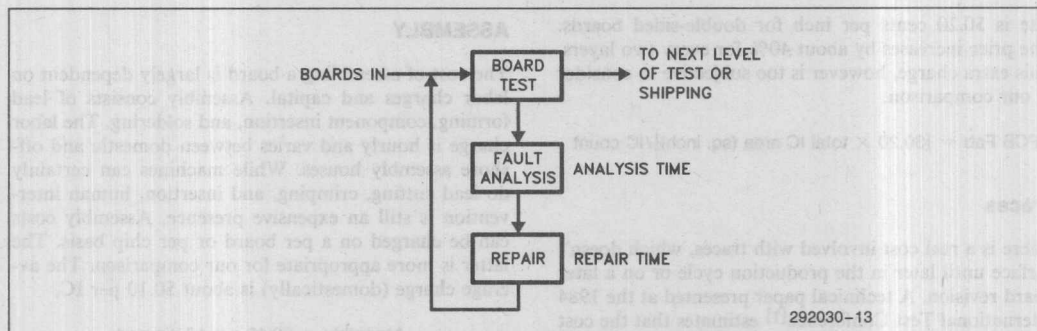


Figure 8. Typical Test and Repair Loop

Errors can occur at any step of the test flow; each time this happens, a test loop is initiated. This loop is depicted in Figure 8. The cost for testing a device depends on the cost of the equipment, depreciation, the labor rate, and other factors that are company dependent. There are several ways to reduce test costs, but the best way is to reduce the probability of errors occurring. There is no question that as the number of ICs increases, so does the probability of error.

With all things considered, the industry reports a nominal test cost of about \$0.15 per IC.<sup>[27][28]</sup>

$$\text{Test cost} = \$0.15 \text{ per 16 pin IC}$$

### REWORK

The cost of rework is best understood by considering the cause of errors in more detail. Errors are typically caused by poor board quality, inadequate solder process, tolerance of insertion, and of course, bad chips. Table 4 shows the average board fault spectrum. The figures are a conclusion reached by EVALUATION ENGINEERING magazine<sup>[10]</sup> as to what the industry is currently seeing. The table shows that the majority of board errors is due to solder shorts. These errors are the result of traces or IC holes being too close, which is what happens on densely populated boards.

Table 4. Average Board Fault Spectrum

Tolerance	20%
Shorts	40%
Insertion	30%
Bad Parts	10%

Of all the material costs associated with rework, the main cost is the time spent on a repair. Considering that it takes approximately two minutes to desolder,

insert, resolder, and clean a component pin<sup>[9]</sup>, one can see that more ICs on a board directly affect cost. Repair times also increase dramatically on multi-layer boards that might have been doubled sided if denser logic was used.

For our comparison, let's assume that our test equipment is 95% efficient in finding solder faults on the first pass (no loop). This leaves 5% of the faults that go undetected and eventually must be found and repaired. The estimated cost per pin based on a \$6.00 hourly wage and the two minute repair time is approximately \$0.02 cents.

$$\text{Rework} = [\$0.02 \times \text{total pin count}] / \text{IC count}$$

It is important to note that the probability of errors is based on a Poisson distribution<sup>[8]</sup> that increases exponentially with the number of pins and components. This distribution is used in wave solder processing to correct for solder errors. Mathematically this is expressed as:

$$P = \frac{e^{-np}(np)^x}{x!}$$

where; P = The probability that a defect will occur  
 n = The number of components  
 p = The fraction defective  
 x = The actual number of defects

This means that the TTL and PAL version of the arbiter have a higher probability of error than the EPLD version. However, to make our comparison easier, let's simplify this to more of a linear relation. For each implementation, the rework cost per IC is calculated by;

$$\text{Rework cost} = [(\text{total pin count}) \times (5\%) \times (\$0.02 \text{ cents})] / \text{IC count}$$

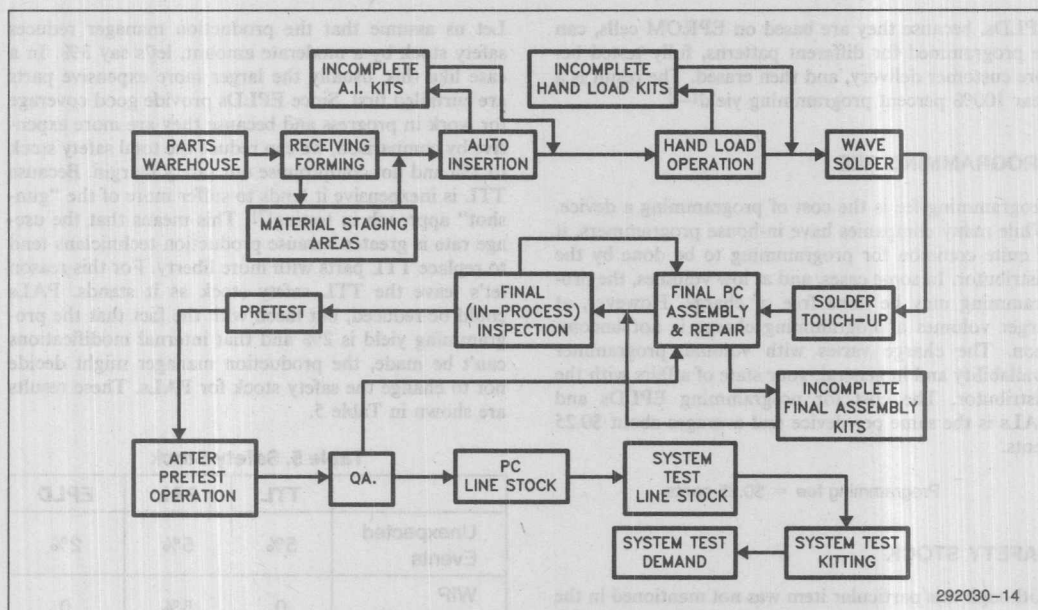


Figure 9. Example of a Production Line

### QUALITY CONTROL

In most production operations, boards go through several steps of quality inspection. The bare board might be inspected after preliminary tests and after system tests. Although 100% inspection should theoretically eliminate all errors, in real life this rarely happens. The main reason for this is the complexity of the production and rework loops as shown in Figure 9.

Quality control's purpose is to remove defective products and either junk them or rework them, neither of which is cost effective. The best approach is to design the quality in, not fix it in. One way to design in quality is by reducing the possibility of errors and increasing the reliability of a product. This is one of the primary advantages of dense logic (like EPLDs and PALs) over TTL.

A survey conducted by *CIRCUITS MANUFACTURING* magazine<sup>[8]</sup> yielded the cost of \$10 to \$50 dollars to inspect, find, and repair a defect on a board. They summarized that the actual cost of inspection is about \$0.004 for each hole on a board. With this in mind, let us assume a 100% inspection of our arbiter circuit for each implementation. This means that each pin (and every trace via) will have to be looked at. The calculation for this is;

$$QC \text{ cost} = (\text{total pin count} \times \$0.004) / \text{IC count}$$

### POWER SUPPLY

Price for 5V, single output, switching power supplies as advertised by several vendors is \$1.00 per watt. The calculation for determining power supply costs in our comparison is:

$$\text{Power cost} = [(5VDC \times I_{CC} \text{ (mA)}) \times \$1.00 \text{ per watt}] / \text{IC count}$$

### Additional Costs

In addition to the more obvious costs, there are several other items that contribute to the "hidden cost" of a system.

### PROGRAMMING LOSS

Because PALs are a one time programmable type of device, full testing can't be done on them without destroying the user's fuses. For this reason PALs have a published programming loss of 2%<sup>[20]</sup>. The cost for this is:

$$\text{Programming loss} = (\text{PAL IC count} \times 0.02) \times \text{PAL cost per IC}$$

EPLDs, because they are based on EPROM cells, can be programmed for different patterns, fully tested before customer delivery, and then erased. The result is a near 100% percent programming yield<sup>[22]</sup>.

### PROGRAMMING FEE

Programming fee is the cost of programming a device. While many companies have in-house programmers, it is quite common for programming to be done by the distributor. In some cases, and at low volumes, the programming may be done free of charge. However, at larger volumes a programming charge is not uncommon. The charge varies with volume, programmer availability and in general, your state of affairs with the distributor. The cost for programming EPLDs and PALs is the same per device and averages about \$0.25 cents.

Programming fee = \$0.25 cents

### SAFETY STOCK

Although this particular item was not mentioned in the inventory section, it plays a very important role in the production world. Safety stock<sup>[21]</sup> is extra ICs ordered to cover for unexpected events. Unexpected here might be a large unforeseen customer order or simply a bad batch of parts.

While industry seems to strive for the optimum JIT (just in time) production<sup>[14][16]</sup>, which stresses minimal inventory until needed, it's not unusual for production managers to carry a five to ten percent inventory buffer depending on the cost of the part. In most cases, the larger expensive parts like microprocessors, peripheral controllers, and other LSI devices are safety stocked in smaller quantities.

Let's assume that the safety stock is to be a maximum of 10%. Five percent might be used to cover for the unexpected occurrences, and five for WIP (work in process) modifications. Since all parts have the same probability of unexpected events we can assign that percentage equally. Justifying the second 5% depends on the IC technology itself. For instance, WIP modifications usually require cuts and jumpers on TTL, therefore it's unnecessary to order the additional 5%. In process modifications to an EPLD are done simply by reprogramming it, here again there is no need for the additional 5%. PALs however cannot be cut and jumpered (internally) nor can they be reprogrammed. Also, there is the possibility that "on the shelf" PALs will be programmed in advance, therefore a WIP mod that impacts their function means that those parts must be obsoleted (junked). In this case, an additional 5% is justifiable.

Let us assume that the production manager reduces safety stock by a moderate amount, let's say 3%. In a case like this, usually the larger more expensive parts are curtailed first. Since EPLDs provide good coverage for work in progress and because they are more expensive by comparison, we can reduce the total safety stock to 2% and not compromise our safety margin. Because TTL is inexpensive it tends to suffer more of the "gun-shot" approach in testing<sup>[7]</sup>. This means that the usage rate is greater because production technicians tend to replace TTL parts with more liberty. For this reason let's leave the TTL safety stock as it stands. PALs could be reduced, but faced with the fact that the programming yield is 2% and that internal modifications can't be made, the production manager might decide not to change the safety stock for PALs. These results are shown in Table 5.

Table 5. Safety Stock

	TTL	PAL	EPLD
Unexpected Events	5%	5%	2%
WIP MODS	0	5%	0
Total	5%	10%	2%

The safety stock calculation for each implementation is:

$$\text{Safety stock} = (\% \text{ of stock} \times \text{IC type} \times \text{IC type cost}) / \text{IC count}$$

### DE-COUPLING CAPACITORS

While adding caps solves many problems due to system noise, it also increases the cost of PCB layout, PCB fab, and adds an additional burden on all of our other costs. For a TTL system, a good de-coupling rule of thumb is to use one 0.01  $\mu\text{f}$  per each synchronous driven gate and at least 0.1  $\mu\text{f}$  per 20 gates regardless of synchronicity. Engineers recognize the need for decoupling and usually take it a step further by using one capacitor per IC. Most boards reflect this practice, which, in itself is very good. However, the addition of all these caps is definitely measurable, in both component and systems cost.

The average cost of a ceramic capacitor in moderate quantities is about half a cent. For our comparison we will follow the accepted practice and de-couple each TTL, PAL, and EPLD device. Our capacitor cost is then:

$$\text{De-coupling cost} = \$0.005 \times \text{IC count}$$

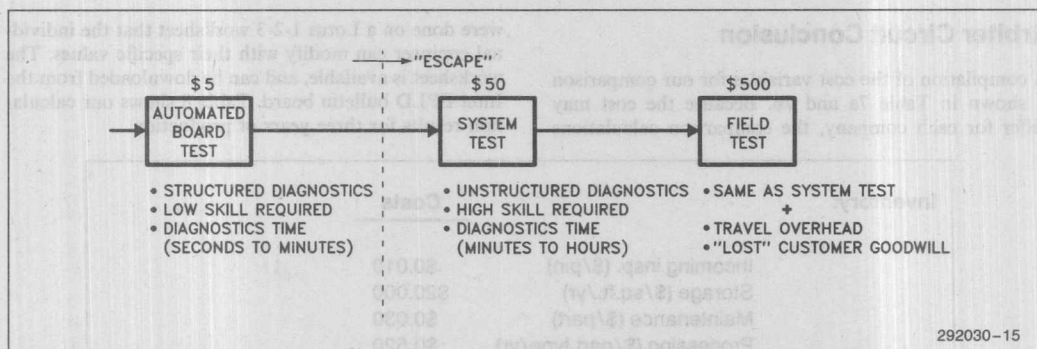


Figure 10. Escape Costs

### Other Costs To Consider

Eventually, some place toward the end of a production line, a board becomes part of a system. At this point it is housed in an enclosure and all the necessary cabling is done. Even here, however, the impact of using a particular IC technology can still be felt.

### DEFECT ESCAPES

One very significant item that the test community acknowledges is the cost of "escapes"[4]. "Escape" is defined as a fault that goes through the early stages of board test undetected. Figure 10 shows the escape relationship. An industry rule of thumb states that the cost to detect a fault increases by an order of magnitude at each stage. This means that if it costs \$5 to find a fault at the board test level, that same fault might cost \$50 at the system level and \$500 at the field level. An important relationship to remember, is that the number of faults per board increases logarithmically, as the number of components on the board increases[6]. The cost of an "escape" is difficult to quantify, but generally, a board with a higher component count has a greater cost[2][8].

### CABLES/WIRING HARNESS

When the number of components or the power requirements of a system are reduced, a reduction in cables and wiring is usually expected. The cost savings here is either in the elimination of cables (because more functions are condensed into an IC) or the reduction of cable gauge or length (because less power is required, in the case of EPLDs). Also, fewer cables means fewer cable ties, connector pins, and mounting hardware. While this is a subjective figure, let's assume that the distributed cost of system cables is \$0.25 per IC.

$$\text{Cable cost} = \$0.25 \times \text{IC count}$$

### ENCLOSURE

Certain applications require reduced packaging or enclosure size. In industrial control for example, each line might require a complete system to monitor its operation. In a case like this, a large bulky box full of boards might not be appropriate. A good example of the benefits that high integration logic provide enclosures, is the third market versions of the popular PC. Many of these companies have fully compatible versions that fit on a single board. EPLDs and PALs are capable of providing a cost savings in this respect. However, while PALs approach the density requirements, their large power needs render them counterproductive to the low power specs of small systems. TTL is just not as effective as either PALs or EPLDs.

For our comparison let us assume the cost of enclosure per chip is \$0.75. The calculation is:

$$\text{Enclosure cost} = \$0.75 \times \text{IC count}$$

Table 6 shows the cable and enclosure costs for the MULTIBUS I circuit. Although the results are based on assumed values, we can see that a larger IC count influences the burdened cost of the system. Our final comparison will not use these figures, but they should be considered.

Table 6. Other Production Costs for Multibus I Circuit

	TTL	PLA	EPLD
Wiring/harness	\$2.750	\$0.500	\$0.250
Enclosure	\$8.250	\$1.500	\$0.750



## Arbiter Circuit Conclusion

A compilation of the cost variables for our comparison is shown in Table 7a and 7b. Because the cost may differ for each company, the comparison calculations

were done on a Lotus 1-2-3 worksheet that the individual engineer can modify with their specific values. The worksheet is available, and can be downloaded from the Intel EPLD bulletin board. Table 8 shows our calculation results for three years of production.

Inventory:		Costs	
Incoming insp. (\$/pin)		\$0.010	
Storage (\$/sq.ft./yr)		\$20.000	
Maintenance (\$/part)		\$0.030	
Processing (\$/part type/yr)		\$0.520	
Safety stock (%)		2%	
Manufacturing:		Costs	
PCB fab. (\$/sq.in.)		\$0.200	
Assembly (\$/part)		\$0.100	
Test (\$/part)		\$0.150	
Rework (\$/pin)		\$0.020	
QC (\$/pin)		\$0.004	
Power (\$/watt)		\$1.000	
Interconn		\$0.020	
Program (\$/part)		\$0.250	
Caps. (each)		\$0.005	
(a)			
Integrated Circuits			
Component Count:			
Package	TTL	PLA	EPLD
DIP14	10		
DIP16	1		
DIP20	0	2	
DIP24			1
		ICs	
		TTL	Types
		PLA	
		EPLD	
			10
			2
			1
Circuit Requirements:		I <sub>CC</sub> (max)	
		Interconnects	
TTL circuit (total mA).		100	36
PLA circuit (total mA).		240	7
EPLD circuit (total mA).		15	0
(b)			

Tables 7a and b. Multibus Arbiter/Controller Cost Variables

AVERAGE COMPONENT COST									
Package	Year 1			Year 2			Year 3		
	TTL	PLA	EPLD	TTL	PLA	EPLD	TTL	PLA	EPLD
DIP14	\$0.25			\$0.20			\$0.19		
DIP16	\$0.35			\$0.30			\$0.27		
DIP20	\$0.55	\$2.00		\$0.38	\$1.70		\$0.35	\$1.56	
DIP24			\$6.00			\$4.20			\$2.90
PRODUCTION COSTS									
Item (costs per part)	Year 1			Year 2			Year 3		
	TTL	PLA	EPLD	TTL	PLA	EPLD	TTL	PLA	EPLD
Components	\$0.259	\$2.000	\$6.000	\$0.209	\$1.700	\$4.200	\$0.197	\$1.560	\$2.900
Incoming Insp.	\$0.142	\$0.200	\$0.240	\$0.142	\$0.200	\$0.240	\$0.142	\$0.200	\$0.240
Inventory									
Maintenance	\$0.027	\$0.038	\$0.045	\$0.027	\$0.038	\$0.045	\$0.027	\$0.038	\$0.045
Storage	\$0.030	\$0.042	\$0.050	\$0.030	\$0.042	\$0.050	\$0.030	\$0.042	\$0.050
Processing	\$0.473	\$0.520	\$0.520	\$0.473	\$0.520	\$0.520	\$0.473	\$0.520	\$0.520
Printed Circuit Board									
Fabrication	\$0.043	\$0.060	\$0.072	\$0.043	\$0.060	\$0.072	\$0.043	\$0.060	\$0.072
Trace costs	\$0.065	\$0.070	\$0.000	\$0.065	\$0.070	\$0.000	\$0.065	\$0.070	\$0.000
Assembly	\$0.089	\$0.125	\$0.150	\$0.089	\$0.125	\$0.150	\$0.089	\$0.125	\$0.150
Board test	\$0.150	\$0.150	\$0.150	\$0.150	\$0.150	\$0.150	\$0.150	\$0.150	\$0.150
Rework	\$0.014	\$0.020	\$0.024	\$0.014	\$0.020	\$0.024	\$0.014	\$0.020	\$0.024
QC	\$0.057	\$0.080	\$0.096	\$0.057	\$0.080	\$0.096	\$0.057	\$0.080	\$0.096
Power Supply	\$0.045	\$0.600	\$0.075	\$0.045	\$0.600	\$0.075	\$0.045	\$0.600	\$0.075
Total Cost/Part	\$1.393	\$3.904	\$7.422	\$1.343	\$3.604	\$5.622	\$1.331	\$3.464	\$4.322
Total Cost/System	\$15.321	\$7.808	\$7.422	\$14.771	\$7.208	\$5.622	\$14.641	\$6.928	\$4.322
Additional Costs/System									
Programming loss	\$0.000	\$0.080	\$0.000	\$0.000	\$0.068	\$0.000	\$0.000	\$0.062	\$0.000
Safety stock	\$0.143	\$0.400	\$0.120	\$0.115	\$0.340	\$0.084	\$0.109	\$0.312	\$0.058
Programming fee	\$0.000	\$0.500	\$0.250	\$0.000	\$0.500	\$0.250	\$0.000	\$0.500	\$0.250
De-coupling caps	\$0.055	\$0.010	\$0.005	\$0.055	\$0.010	\$0.005	\$0.055	\$0.010	\$0.005
True mfg. cost/system	\$15.518	\$8.798	\$7.797	\$14.941	\$8.126	\$5.961	\$14.804	\$7.813	\$4.635

The comparison in component costs shows that the EPLD costs more than either a TTL or PAL IC. As costs are added, the figures for TTL and PALs begin to approach the cost of an EPLD. These are shown on the line labeled "Total cost/part".

The "Total cost/system" line shows the actual cost when all the ICs are considered. For the first year, the TTL version is the more expensive implementation, and the EPLD numbers look very favorable.

The "True mfg. cost/system" line results after additional costs are figured in. Here we see that the first year, the EPLD version already provides a \$1 savings over the PAL version, and that the cost of the TTL implementation is very high. Also, the inserted cost per IC at this point is, \$1.15 for TTL, \$2.40 for PAL and \$1.80 for the EPLD. This is in line with the inserted costs that we mentioned earlier.

The production costs for two additional years shows that the decreasing price of EPLDs (based on the curve of Figure 5) will continue to provide costs savings as production ramps up in quantities.

In terms of functional benefits, the EPLD implementation is the most beneficial because;

- The chip count has gone down, one EPLD has replaced 11 TTL ICs in one implementation, and 2 PALs in the other, reducing the cost and time of:
  - board layout
  - board fab
  - assembly
  - rework
- The reliability of the board has increased. Fewer components translates into less probability of error.
- Modifications are easier to make. Instead of cuts and jumpers (for TTL), or throwing away a PAL, a change is re-programmed.
- The need for de-coupling caps is reduced. All those individual ICs are eliminated and in some cases the distributed capacitance of the board may be enough de-coupling.
- Power supply requirements are small. The active current requirements are much smaller with EPLDs. This in turn reduces the need for large power supplies and fans.
- Cable requirements and enclosure benefits have been improved. Since EPLDs provide better integration over TTL and PALs, the size of the system will be smaller. This translates into fewer boards and cables.
- Inventory is reduced. One EPLD replaces many TTL devices. Also, "on the shelf" programmed EPLDs can be reused in a pinch, PALs can't.

Less expense and probability of "escapes". The time and cost of finding and fixing escape problems is re-

duced to one reprogrammable IC. In the field, this translates into less "down time" for the customer and a higher level of customer "goodwill" for the OEM.

Allows capability for customized hardware. Specific customer requirements can be implemented. Also, DIP switches and configuration jumpers may not be necessary in many cases, since configurations can be programmed into the EPLD.

## Development Costs

As mentioned earlier, the costs of development are usually dismissed as NRE. One reason for this is the difficulty in pegging down these costs. However, while money might be expendable at this stage, time is usually critical. Time saved at the front end can make a difference in beating the competition to market. The following topics are presented for consideration. No costs are assigned to them.

## RESEARCH

The amount of time spent researching components, component sources, and technical data can be very large. Designs done with a large IC count require more research and analysis time. Higher integration devices require learning curve time, but, in the long run this tends to reduce research time, especially in future designs.

## PROTOTYPING

For most companies, prototypes are three to five level wire wrap boards built by inhouse technicians or outside contractors. During prototype fab, a certain amount of work has to be done to each IC. Part of this work is, adding bypass caps, labeling chips, and lead forming. In smaller companies, the board might be hand wrapped. Larger companies might use an automatic wrapper. Once the board is wrapped, a continuity check is done on each wire net to insure connections and minimize shorts.

The turn around time for a protoboard is one to two weeks and can be shortened by paying a premium price. An alternate way of shortening this time is to simplify the board by using denser ICs.

## DEBUGGING

Fixing bugs on a protoboard involves unwrapping and wrapping connections, as well as replacing ICs. Making mods on a TTL board is very time consuming and error prone due to the large numbers of wires. Making mods with PALs is expensive since the part usually has to be junked. EPLDs in contrast, are re-programmable and lend themselves to all the revisions that are common in the early design stages.

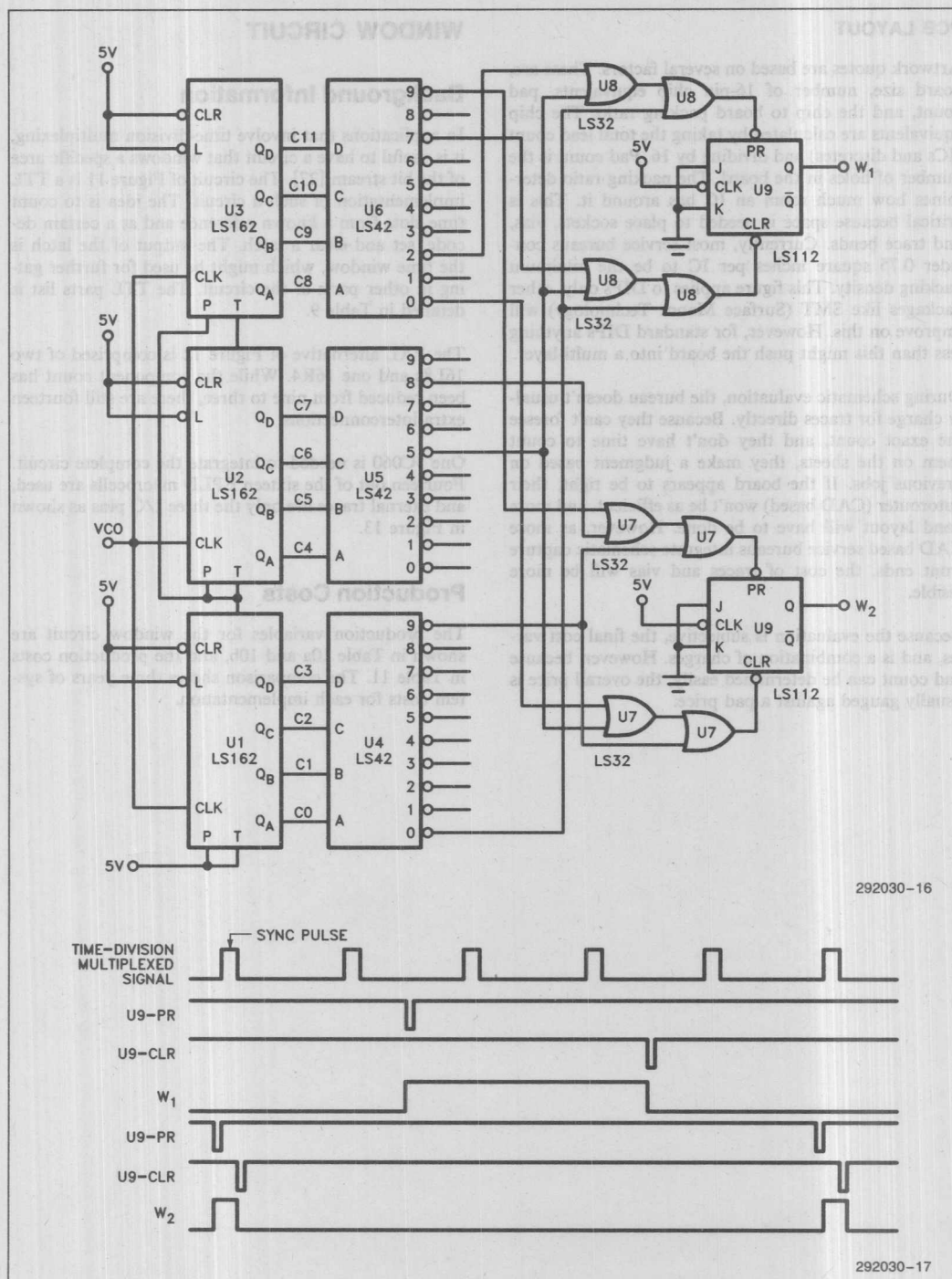


Figure 11. Time Window Generator, TTL Circuit



## PCB LAYOUT

Artwork quotes are based on several factors. These are, board size, number of 16-pin chip equivalents, pad count, and the chip to board packing ratio. The chip equivalents are calculated by taking the total lead count (ICs and discretes) and dividing by 16. Pad count is the number of holes in the board. The packing ratio determines how much room an IC has around it. This is critical because space is needed to place sockets, vias, and trace bends. Currently, most service bureaus consider 0.75 square inches per IC to be the minimum packing density. This figure applies to DIPs only, other packages like SMT (Surface Mount Technology) will improve on this. However, for standard DIPs anything less than this might push the board into a multi-layer.

During schematic evaluation, the bureau doesn't usually charge for traces directly. Because they can't foresee the exact count, and they don't have time to count them on the sheets, they make a judgment based on previous jobs. If the board appears to be tight, their autorouter (CAD based) won't be as efficient, and more hand layout will have to be done. However, as more CAD based service bureaus integrate schematic capture front ends, the cost of traces and vias will be more visible.

Because the evaluation is subjective, the final cost varies, and is a combination of charges. However, because pad count can be determined easily, the overall price is usually gauged against a pad price.

## WINDOW CIRCUIT

## Background Information

In applications that involve time-division multiplexing, it is useful to have a circuit that windows a specific area of the bit stream [27]. The circuit of Figure 11 is a TTL implementation of such a circuit. The idea is to count time slots from a known reference and at a certain decode, set and clear a latch. The output of the latch is the time window, which might be used for further gating in other parts of the circuit. The TTL parts list is detailed in Table 9.

The PAL alternative of Figure 12 is comprised of two 16L8s and one 16R4. While the component count has been reduced from nine to three, there are still fourteen extra interconnections.

One 5C060 is needed to integrate the complete circuit. Fourteen out of the sixteen EPLD macrocells are used, and external traces are only the three I/O pins as shown in Figure 13.

## Production Costs

The production variables for the window circuit are shown in Table 10a and 10b, and the production costs in Table 11. The comparison shows three years of system costs for each implementation.

			AP 307	Area(in <sup>2</sup> )	\$
U1	LS162	16	32	.24	.49
U2	LS162	16	32	.24	.49
U3	LS162	16	32	.24	.49
U4	LS42	16	13	.24	.39
U5	LS42	16	13	.24	.39
U6	LS42	14	13	.24	.39
U7	LS32	14	9.8	.21	.18
U8	LS32	14	9.8	.21	.18
U9	LS112	14	6	.21	.29

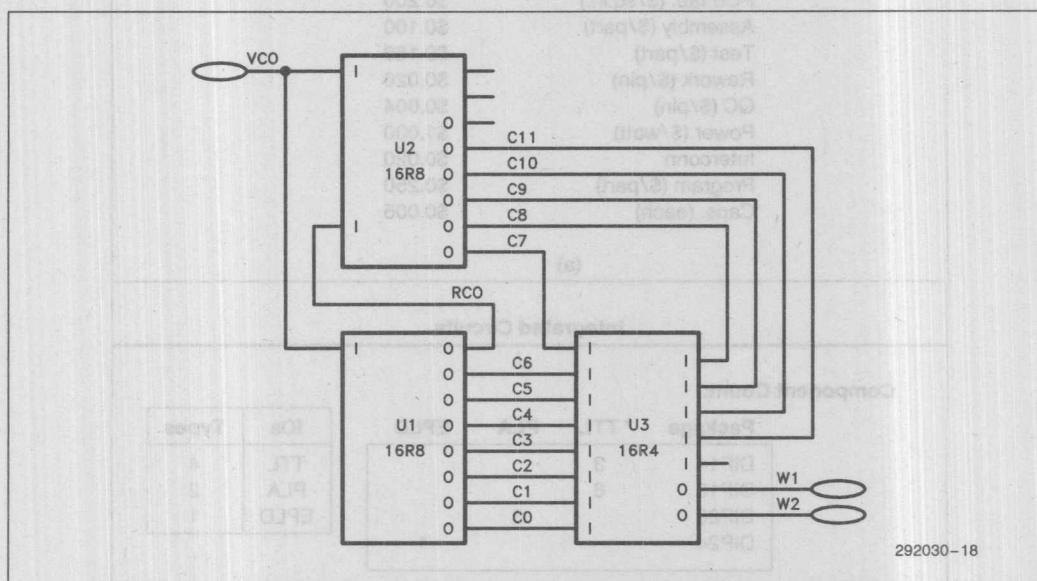


Figure 12. Time Window Generator, PAL Circuit

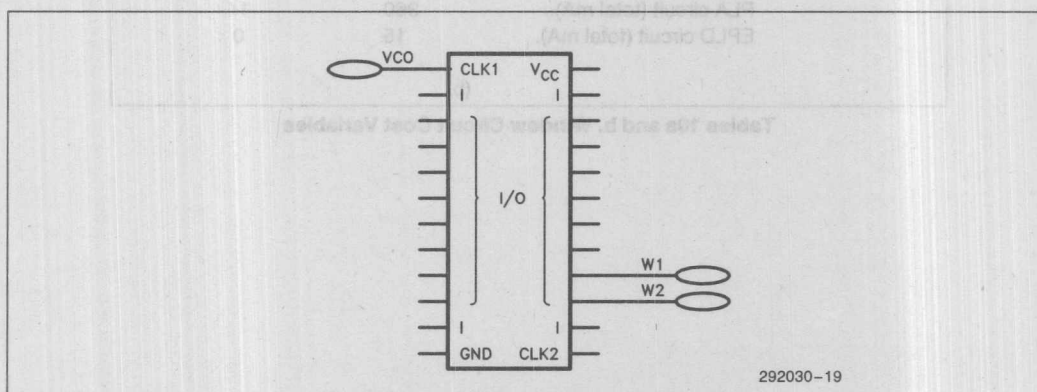


Figure 13. Time Window Generator, EPLD Circuit

<b>Inventory:</b>		<b>Costs</b>	
	Incoming insp. (\$/pin)		\$0.010
	Storage (\$/sq.ft./yr)		\$20.000
	Maintenance (\$/part)		\$0.030
	Processing (\$/part type/yr)		\$0.520
	Safety stock (%)		2%
<b>Manufacturing:</b>		<b>Costs</b>	
	PCB fab. (\$/sq.in.)		\$0.200
	Assembly (\$/part)		\$0.100
	Test (\$/part)		\$0.150
	Rework (\$/pin)		\$0.020
	QC (\$/pin)		\$0.004
	Power (\$/watt)		\$1.000
	Interconn		\$0.020
	Program (\$/part)		\$0.250
	Caps. (each)		\$0.005
(a)			
<b>Integrated Circuits</b>			
<b>Component Count:</b>			
<b>Package</b>	<b>TTL</b>	<b>PLA</b>	<b>EPLD</b>
DIP14	3		
DIP16	6		
DIP20		3	
DIP24			1
<b>Circuit Requirements:</b>		<b>I<sub>CC</sub> (max)</b>	<b>Interconnects</b>
TTL circuit (total mA).		160	52
PLA circuit (total mA).		360	14
EPLD circuit (total mA).		15	0
(b)			

Tables 10a and b. Window Circuit Cost Variables

Table 11. Window Circuit Production Costs

AVERAGE COMPONENT COST									
Package	Year 1			Year 2			Year 3		
	TTL	PLA	EPLD	TTL	PLA	EPLD	TTL	PLA	EPLD
DIP14	\$0.22			\$0.19			\$0.17		
DIP16	\$0.44			\$0.37			\$0.26		
DIP20		\$2.00			\$1.70			\$1.56	
DIP24			\$6.00			\$4.20			\$2.90
PRODUCTION COSTS									
Item (costs per part)	Year 1			Year 2			Year 3		
	TTL	PLA	EPLD	TTL	PLA	EPLD	TTL	PLA	EPLD
Components	\$0.367	\$2.000	\$6.000	\$0.310	\$1.700	\$4.200	\$0.230	\$1.560	\$2.900
Incoming Insp.	\$0.153	\$0.200	\$0.240	\$0.153	\$0.200	\$0.240	\$0.153	\$0.200	\$0.240
Inventory									
Maintenance	\$0.029	\$0.038	\$0.045	\$0.029	\$0.038	\$0.045	\$0.029	\$0.038	\$0.045
Storage	\$0.032	\$0.042	\$0.050	\$0.032	\$0.042	\$0.050	\$0.032	\$0.042	\$0.050
Processing	\$0.231	\$0.347	\$0.520	\$0.231	\$0.347	\$0.520	\$0.231	\$0.347	\$0.520
Printed Circuit Board									
Fabrication	\$0.046	\$0.060	\$0.072	\$0.046	\$0.060	\$0.072	\$0.046	\$0.060	\$0.072
Trace costs	\$0.116	\$0.093	\$0.000	\$0.116	\$0.093	\$0.000	\$0.116	\$0.093	\$0.000
Assembly	\$0.096	\$0.125	\$0.150	\$0.096	\$0.125	\$0.150	\$0.096	\$0.125	\$0.150
Board test	\$0.150	\$0.150	\$0.150	\$0.150	\$0.150	\$0.150	\$0.150	\$0.150	\$0.150
Rework	\$0.015	\$0.020	\$0.024	\$0.015	\$0.020	\$0.024	\$0.015	\$0.020	\$0.024
QC	\$0.061	\$0.080	\$0.096	\$0.061	\$0.080	\$0.096	\$0.061	\$0.080	\$0.096
Power Supply	\$0.089	\$0.600	\$0.075	\$0.089	\$0.600	\$0.075	\$0.089	\$0.600	\$0.075
Total Cost/Part	\$1.385	\$3.754	\$7.422	\$1.328	\$3.454	\$5.622	\$1.248	\$3.314	\$4.322
Total Cost/System	\$12.463	\$11.263	\$7.422	\$11.953	\$10.363	\$5.622	\$11.233	\$9.943	\$4.322
Additional Costs/System									
Programming loss	\$0.000	\$0.120	\$0.000	\$0.000	\$0.102	\$0.000	\$0.000	\$0.094	\$0.000
Safety stock	\$0.165	\$0.600	\$0.120	\$0.140	\$0.510	\$0.084	\$0.104	\$0.468	\$0.058
Programming fee	\$0.000	\$0.750	\$0.250	\$0.000	\$0.750	\$0.250	\$0.000	\$0.750	\$0.250
De-coupling caps	\$0.045	\$0.015	\$0.005	\$0.045	\$0.015	\$0.005	\$0.045	\$0.015	\$0.005
True mfg. cost/system	\$12.673	\$12.748	\$7.797	\$12.137	\$11.740	\$5.961	\$11.381	\$11.269	\$4.635



The production costs again show that the system cost for the first year is better with EPLDs. The two consecutive years show that the declining price of EPLDs make them an excellent candidate for systems that will ramp up production at that time.

### Window Circuit Conclusion

The TTL version of the circuit was implemented with MSI counters and decoders. As a result, the PAL implementation was bound by the number of count bits and had to be programmed into two PALs. In circuits like this, it is useful to rewire the decode for different counts depending on the application. The PAL implementation allows this by incorporating the decode and output latches into one IC.

The EPLD implementation tackles the MSI integration quite easily and also provides the capability to reprogram the decoder. Since the counter and output latches consist of fourteen registered outputs, the sixteen macrocells of the 5C060 easily accommodate the needed functions.

### SUMMARY

We have examined the hidden costs of production and how they differ for several logic alternatives. By examining these costs, we have shown that while an EPLD is presently a more expensive part, its level of integration reduces system costs and improves reliability. The following items should be considered when evaluating logic alternatives:

- system cost is determined by more than component cost
- system cost and reliability is influenced by the type and amount of components used
- semiconductors have a life cycle that determines their present price at design, and at production time

In summary, when all system costs are considered, EPLDs can provide cost savings to the design and production of most board designs.

### REFERENCES

1. The Future Is Now: Extending CAE into test of custom VLSI.  
Robert S. Broughton, Tektronix.  
Michael G. Brashler, Tektronix.  
IEEE International Test Conference Proceedings, 1984
2. Reducing The Cost of Quality Through Test Data Management.  
Paul N. Manikas, GenRad Inc.  
Stephen G. Eichenlaub, Harvard University.  
IEEE International Test Conference Proceedings, 1983
3. A Quantitative Analysis Of The Trade-offs Between Higher Capital Investment and Higher Yield In PCB Testing.  
Mark A. Myers, Teradyne Inc.  
IEEE International Test Conference Proceedings, 1984
4. An Analysis Of The Cost And Quality Impact Of LSI/VLSI Technology On PCB Test Strategies.  
Mark A. Myers, Teradyne Inc.  
IEEE International Test Conference Proceedings, 1983
5. IC Quality Control By The User.  
Roger Dunn, Xerox Corp.  
IEEE International Test Conference Proceedings, 1983
6. An Analysis Of The Economics of Self Test.  
P. Varma, University of Manchester.  
A. P. Ambler, University of Manchester.  
K. Baker, GEC Research Labs.  
IEEE International Test Conference Proceedings, 1984
7. In Circuit Testability Factors: Shoot With A Rifle.  
Douglas W. Raymond, Zehntel Production Services.  
IEEE International Test Conference Proceedings, 1984
8. Seven Steps To Zero Defects.  
D. W. Rudd, AT&T Technologies.  
Circuits Manufacturing, June 1986
9. Rework Forum  
Donald Ford, Senior Editor.  
Circuits Manufacturing, September 1986
10. Manufacturing Defect Analyzers: Annual Round-up.  
Evaluation Engineering magazine, August 1986
11. Assembly: Automation Makes It Better.  
Roland W. Roy and Gordon Weeks, Andover Controls.  
Circuits Manufacturing, February 1986
12. Shrinking Lines Squeeze Processes.  
Jerry Murray, West Coast Editor.  
Circuits Manufacturing, September 1986
13. Ribbon Cable for Reliable Interconnections.  
Bennett W. Brachman, Xport Trading Inc.  
Electronic Packaging and Production magazine, July 1986

14. TQC and JIT: Partners In Production.  
Rick Walleigh, Hewlett Packard.  
Circuits Manufacturing, February 1986
15. Automated Handling/Sorting: Multisite Development Moves to Back Burner.  
Evaluation Engineering magazine, May 1986
16. Software Charts The Course of Component Testing.  
Ronald Pound, Editor.  
Electronic Packaging and Production magazine, June 1986
17. Complexity, PLDs Drive The Market.  
Evaluation Engineering magazine, July 1986
18. Intel User Defined Logic Handbook.  
Intel Corp. 1986
19. VLSI Semicustom Design Guide.  
CMP Publications, Summer 1986
20. AMD Programmable Array Logic Handbook.  
Advanced Micro Devices, 1984
21. Handbook Of Industrial Engineering.  
Gavriel Salvendy, Editor, Purdue University  
John Wiley & Sons Publications
22. Components Quality/Reliability Handbook.  
Intel Corporation.
23. The Cost Edge.  
DM DATA Corp.  
Scottsdale, AZ
24. Semiconductor Purchasing Strategies Integrated Circuits Engineering Corp.  
Scottsdale, AZ
25. Status 1986 Integrated Circuits Engineering Corp.  
Scottsdale, AZ
26. EDN Semicustom Design Series  
EDN Magazine, 1985
27. EDN Design Ideas  
EDN Magazine, 1985

November 1992

# Metastability Characteristics of Intel EPLDs

**THOM BOWNS**  
PROGRAMMABLE LOGIC APPLICATIONS  
INTEL CORPORATION

Order Number: 292071-003

WHAT METASTABILITY IS .....	3-228
HOW METASTABILITY AFFECTS SYSTEM DESIGN .....	3-228
HOW METASTABILITY IS QUANTIFIED .....	3-230
METASTABILITY DATA FOR INTEL EPLDS .....	3-232

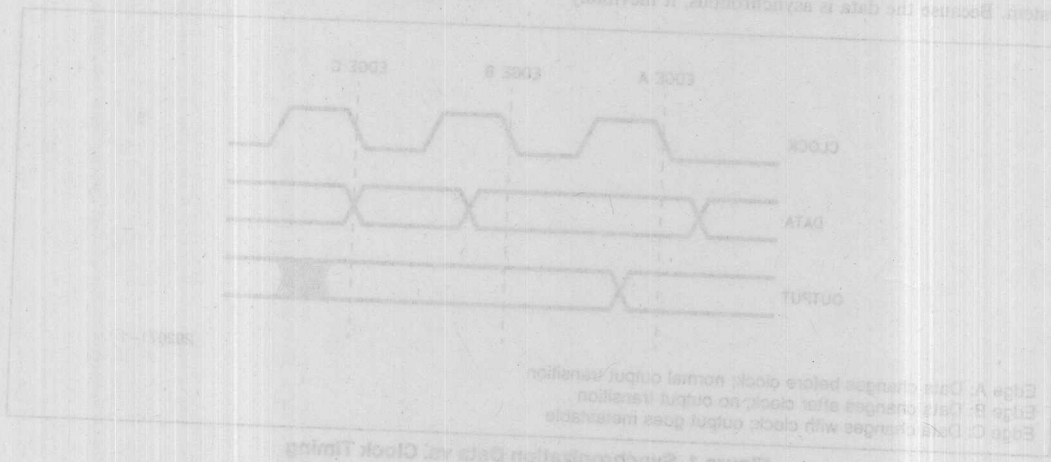
HOW INTEL EPLDS COMPARE AGAINST OTHER PLDS .....	3-232
SUMMARY .....	3-233
REFERENCES .....	3-233
ACKNOWLEDGEMENTS .....	3-233

When one of these registers enters a metastable state, the voltage at the output pin can be seen to oscillate or hover in between HIGH and LOW. Intel CMOS EPLDs have high gain buffers between the register circuitry and the output pin. The output pins will always be at one rail or the other and therefore will never show an in-between or oscillatory state. Instead, they manifest metastability by a state transition which maintains the output waveform approximately later than allowed by the normal clock to output delay (T<sub>CO</sub>) specification of the device. To guard against the effects of metastability, one designs the system to wait a sufficient time after the register. This additional wait time after T<sub>CO</sub> is called T<sub>MET</sub>. The designer uses the device's specified metastability transition character as a Mean Time Between Failures (MTBF) value that means the target system requirements and then calculates a T<sub>MET</sub> based upon that MTBF.

HIGH and a LOW for a prolonged period. This can occur when the register's setup or hold parameters are violated, but it won't data input to the register transition for some to the clock edge. Figure 1 shows the setup conditions of when data may arrive with reference to the system clock, and when metastability may occur.

## HOW METASTABILITY AFFECTS SYSTEM DESIGN

Most systems have the capability to read input data from an asynchronous source such as a keyboard or a modem. They are called asynchronous because the timing of the data from these sources is not in synchronization with the system clock. This type of data must first be synchronized before the system can use it. This is usually done by placing a synchronization register stage between the asynchronous data source and the rest of the system. Because the data is asynchronous, it inevitably





## WHAT METASTABILITY IS

Edge triggered registers (otherwise known as "flip-flops") are designed to propagate and store data applied to their "D" inputs. If the data applied is HIGH just before the clock edge, the Q output transitions to HIGH. If the data applied is LOW just before the clock edge, the Q output transitions to LOW. These are the two normal states for a register. However, registers may be forced into a "metastable" state, in which the output either oscillates or simply remains in between a HIGH and a LOW for a prolonged period. This can occur when the register's setup or hold parameters are violated; that is, when data input to the register transitions too close to the clock edge. Figure 1 shows the three possibilities of when data may arrive with reference to the system clock, and when metastability may occur.

## HOW METASTABILITY AFFECTS SYSTEM DESIGN

Most systems have the capability to read input data from an asynchronous source such as a keyboard or a modem. They are called asynchronous because the timing of the data from these sources is not in synchronization with the system clock. This type of data must first be synchronized before the system can use it; this usually means placing a synchronization register stage between the asynchronous data source and the rest of the system. Because the data is asynchronous, it inevitably

violates setup and hold requirements of the synchronization register at some time during normal operation and drives the register into a metastable state. Once in the metastable state, the synchronization register eventually resolves to either a HIGH or LOW. How easy the register can be driven into, and how long it remains in the metastable state depends upon noise, ambient conditions, and the register's process technology. For example, FAST\* TTL is more immune to and resolves more quickly from a metastable state than LS TTL.

In some older technologies, register outputs are fed directly to the device pin with no intermediate amplification stage. When one of these registers enters a metastable state, the voltage at the output pin can be seen to oscillate or hover in between HIGH and LOW. Intel CMOS EPLDs have high gain buffers between the register circuitry and the output pin. The output pins will always be at one rail or the other, and therefore will never show an in-between or oscillatory state. Instead, they manifest metastability by a late transition: when metastable, the output transitions appreciably later than allowed by the normal clock to output delay ( $T_{CO}$ ) specification of the device. To guard against the effects of metastability, one designs the system to wait a sufficient time after the register's specified  $T_{CO}$  to ensure valid data from the register. This additional wait time after  $T_{CO}$  is called  $T_{MET}$ . The designer uses the device's supplied metastability characteristics, chooses a Mean Time Between Failures (MTBF) value that meets the target system requirements, and then calculates a  $T_{MET}$  based upon that MTBF.

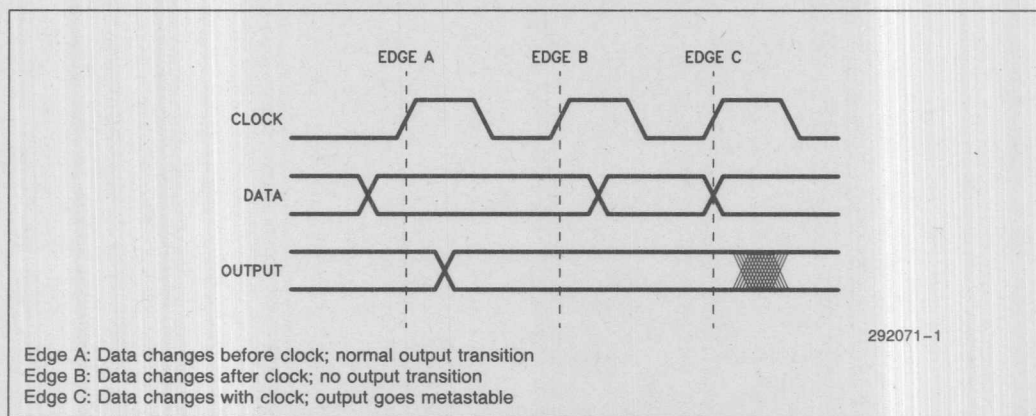


Figure 1. Synchronization Data vs. Clock Timing

\*FAST is a trademark of Fairchild Semiconductor Corporation.

Since metastability is a probabilistic quantity, one cannot simply say that after a given  $T_{MET}$  the register is absolutely guaranteed to have resolved, but rather we refer to the probability that the register will have resolved. Further, since MTBF is exponential in nature, the failure curve is greatly skewed toward early failures. For instance, say a given register claims an MTBF of 100 years for a given  $T_{MET}$ . Intuitively, we would expect a 50% probability of a failure in the first 100 years. However, the probability is actually 63.2% that the register will manifest a resolution failure in the first 100 years. Equation 1 gives a very close approximation of the probability of failure based on the MTBF of each synchronizer, the number of systems in use, the number of synchronizers on each system, the length of time in service, and the number of failures allowable in that time. As an example, given ten systems that each have five synchronizers with MTBFs of 1000 years, what is the possibility of one failure in five years? Using equation 1 and solving for P (Probability):

**Equation 1**

$$P = 1 - (e^{-(T/MTBF)})^{(U/F)}$$

$$P = 1 - (e^{-(5/1000)})^{(50/1)}$$

$$P = 1 - (e^{-0.005})^{50}$$

$$P = 1 - (0.995)^{50}$$

$$P = 1 - (0.775)$$

$$P = 0.221 = 22.1\%$$

Where T = time before failure in years, U = number of synchronizer units operating, and F = the number of failures expected.

From this equation we can see that the intuitive concept of MTBF is inadequate, and that an accurate

failure prediction algorithm is necessary. The concept of MTBF is very complicated and may seem unnecessarily technical, but it is essential to know what you're getting when a PLD manufacturer claims 100 years MTBF; that's for one synchronizer alone. Adding more synchronizers increases the probability of failure that much more.

Figure 2 shows the simplified diagram of a synchronization register feeding the rest of the system, which is shown here as simply another register. The minimum clock period for a single stage synchronization scheme such as this is given by equation 2. The clock period cannot be any shorter than the maximum TCO of the synchronization register plus the maximum TSU of the system plus  $T_{MET}$  for the given system conditions.

Equation 2 solves for the margin afforded by a given clock frequency.

**Equation 2**

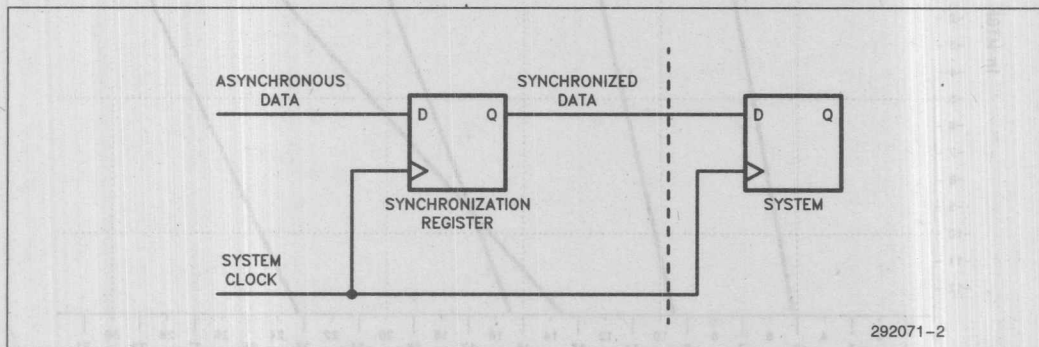
$$\text{Margin} = 1/F_C - (T_{SU} + T_{CO} + T_{MET})$$

where  $F_C$  is the system clock frequency,  $T_{SU}$  is the system's input data setup to clock edge minimum time,  $T_{CO}$  is the synchronization register's clock to output delay, and  $T_{MET}$  is the additional time after  $T_{CO}$  that the designer is willing to wait. Equation 3 allows the designer to determine what sort of  $T_{MET}$  he can expect for a chosen MTBF.

**Equation 3**

$$T_{MET} = (\tau \cdot \ln(T_W \cdot F_C \cdot F_D \cdot MTBF)) - T_{CO}$$

where  $\tau$  and  $T_W$  are the metastability characteristic constants for the selected synchronization register type, MTBF is the mean time between failures expected,  $F_C$  is the system's clock frequency, and  $F_D$  is the average



**Figure 2. Single-Stage Synchronizer**

MTBF when  $T_{MET}$  is chosen and the other factors are known.

#### Equation 4

$$MTBF = \frac{e^{\left(\frac{T_{MET} + T_{CO}}{\tau}\right)}}{T_W \cdot F_C \cdot F_D}$$

It is important to note that each register within an Intel EPLD is independent; if one register should go metastable, other registers within the device are not affected. Thus if several macrocell or input registers are used in parallel as a byte-wise synchronizer array, one need not be concerned about one register affecting the data in another register. The only problem that may arise would be if the incoming data bytes had some timing skew between the bits that make it up. If one input data bit transitioned later than the others, driving it's synchronizer into metastability, the worst case scenario would be that the system would sample the synchronizers before that register settled. That would mean that the data at that point would most likely be invalid, and the next sampled data byte might also be invalid. However, with a sufficient  $T_{MET}$  to allow for metastable settling, the probability of this occurrence can be reduced to a manageable level.

Each device has two metastability characteristic constants,  $\tau$  (or "Tau") and  $T_W$ , which are used to determine the probability of metastable occurrences. As shown in Figure 3, this probability decreases at an exponential rate, according to the value of  $\tau$ .  $T_W$  defines the likelihood that the register will enter the metastable state in the first place; it is also known as the "failure window". Given  $\tau$  and  $T_W$ ,  $T_{MET}$  can be calculated for a chosen MTBF, according to equation 3.

The metastability constants are determined by running the register under test through billions of clock cycles with randomly changing input data. For each increment of  $T_{MET}$ , the number of late transitions are counted within a given time period. A late transition is when the delay from the clock edge until when the register's output changes takes longer than it's observed  $T_{CO}$  allows; we interpret that as an occurrence of a metastable event. The number of failures recorded for each increment of  $T_{MET}$  is logged and then run through the following equations, producing  $\tau$  and  $T_W$ .

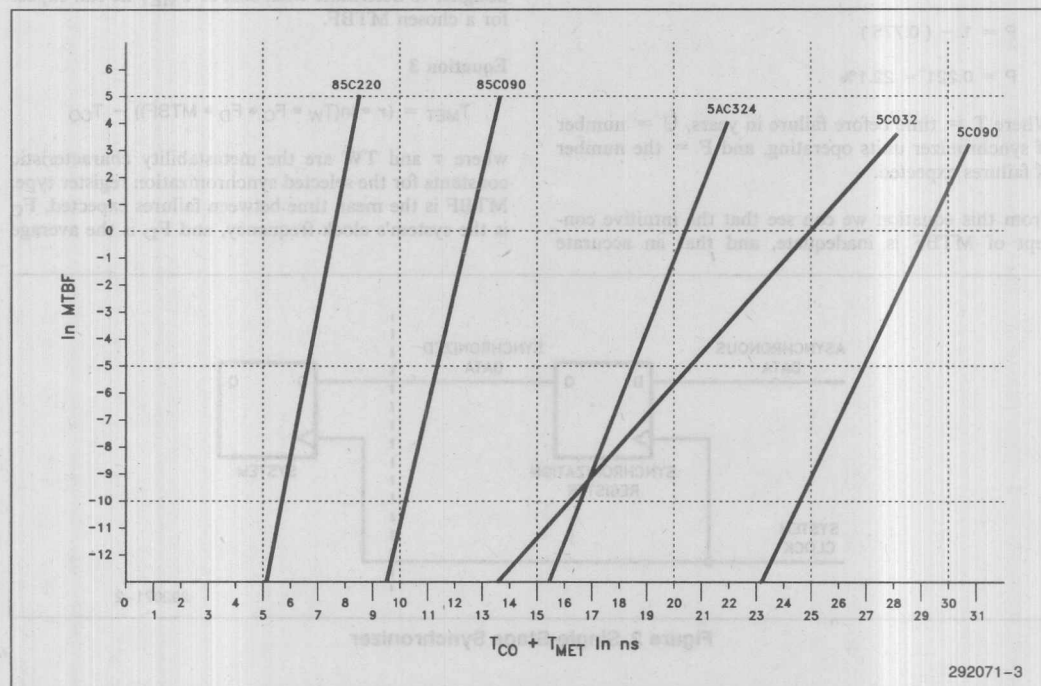


Figure 3.  $\tau$  Curves for Intel EPLD Families

## Equation 5

$$\tau = \frac{n \sum_{i=1}^n X_i^2 - \left( \sum_{i=1}^n X_i \right)^2}{n \sum_{i=1}^n X_i Y_i - \left( \sum_{i=1}^n X_i \right) \left( \sum_{i=1}^n Y_i \right)}$$

## Equation 6

$$T_W = \sum_{i=1}^n \left( \frac{N_i \cdot e^{\left( \frac{T_{CO} + T_{MET}}{\tau} \right)}}{N_{C_i} \cdot F_C} \right) \cdot \frac{1}{n}$$

The X axis is scaled as the  $T_{CO} + T_{MET}$  value in nanoseconds, and the Y axis is the Natural logarithm of the number of seconds between failures (actual number of failures recorded divided into the amount of time taken to accumulate those failures).  $X_i$  and  $Y_i$  are the delay value and log of the failures, respectively, at point  $i$ .  $N_i$  is the number of events recorded at point  $i$ , and  $N_{C_i}$  is the number of clock pulses which occurred within that time.

When plotted on semi-log paper, the resulting line shows the  $\tau$  slope. The Y-axis intercept point is related to  $T_W$ .

Metastable failure characteristics for each EPLD are ascertained by using the setup in Figure 4. The register is clocked by a 2MHz, 25% duty cycle pulse, and register data is provided via a 1MHz, 50% duty cycle pulse. The data is free running with respect to the clock to afford truly random data distribution with the base period of about 1 transition per clock cycle. Since the data transitions are evenly distributed over the entire clock

period, we may assume that the data transitions are also evenly distributed within the failure window (violating setup and hold). The register under test or DUT (device under test) clock is provided by the first output of a dual pulse generator. The second output is delayed by  $T_{CO} + T_{MET}$  and fed into the metastable characterization unit. The output of the DUT is thus sampled at  $T_{CO} + T_{MET}$ , and it is sampled again approximately 100 ns later. If the two samples disagree, the DUT must have transitioned after  $T_{CO} + T_{MET}$ , and therefore must have been metastable. For each metastable event detected, a pulse is generated to increment the event counter. In about half of the cases, the DUT may enter the metastable state and resolve to the same logic level as before, and so not give an external indication of metastability. However, since cases like this will not affect system performance, they are not figured into the MTBF equations.

To run the test,  $T_{CO}$  is characterized for ambient conditions and used as a base delay. The first test begins, and the counter catches  $X$  metastable failure events for 60 seconds. The number of metastable failures is recorded for that particular  $T_{MET}$ .  $T_{MET}$  is increased by .2 nS and another  $X$  events are counted for 60 seconds. This is continued until the events get spaced out to where less than 100 events occur in 60 seconds, and the remaining data points are arrived at by counting 100 events for  $X$  seconds and recording the results appropriately. The results are plotted on semi-log paper, and the slope of the resultant curve is  $\tau$ . Since the raw data points will most likely not fall along a perfectly straight line, the only way to get an accurate value for  $\tau$  is to either draw a best-fit line through the data points and physically measure the slope, or to run the data through equations 5 and 6, given above.

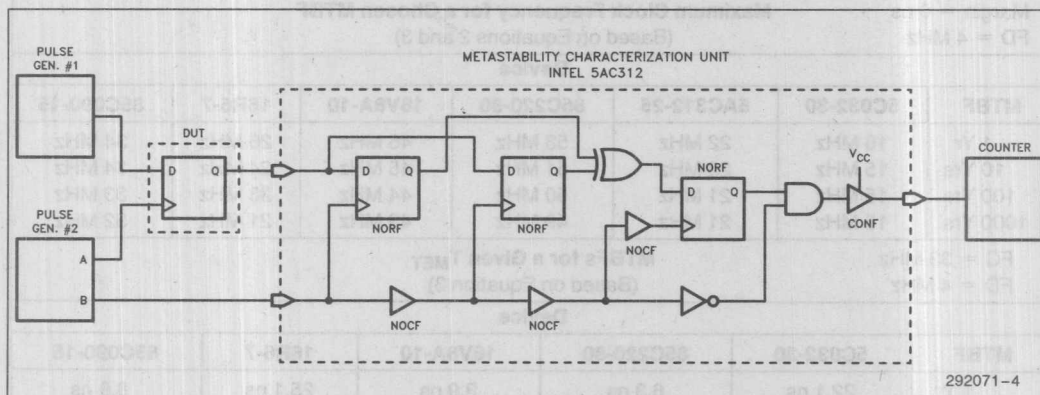


Figure 4. Metastability Characterization Setup



## METASTABILITY DATA FOR INTEL EPLDS

Intel EPLDs are based upon fast, CMOS EPROM process technologies which make them inherently resilient against metastability. The metastability characteristics for each device are dependent upon its design and process technology, thus all Intel EPLDs within a given family have similar metastable characteristics.

The metastability characteristic constants for each Intel EPLD are shown in Table 1. Corresponding metastability characteristic curves for each family are shown in Figure 3.

## HOW INTEL EPLDS COMPARE AGAINST OTHER PLDS

The first half of Table 2 lists maximum allowable clock frequencies for a single-stage synchronization scheme for various EPLDs. This determination comes from a derivation of equations 2 and 3, with the margin set for 0 nanosecond. The other factors are calculated to give a resulting maximum clock frequency. The second half of Table 2 shows the  $T_{MET}$  values that achieve different MTBF levels for several PLD technologies. The metastability characteristics of the other PLDs were determined empirically using the same bench testing method as the Intel EPLDs. It is easy to see that Intel EPLDs have very good metastable resolution characteristics when compared against other technologies.

Table 1. Intel EPLD Metastability Characteristic Constants

Device	$\tau$	$T_w$
5C031	0.82 ns	$5.8 \times 10^2$
5C032	0.86 ns	$2.4 \times 10^{-2}$
5C060	0.90 ns	$4.0 \times 10^0$
5C090	0.44 ns	$2.0 \times 10^{14}$
5AC312	0.42 ns	$1.4 \times 10^4$
5AC324	0.35 ns	$8.6 \times 10^{10}$
5AC324 (Input Registers)	0.39 ns	$3.0 \times 10^{19}$
85C220	0.22 ns	$5.2 \times 10^1$
85C224	0.24 ns	$4.4 \times 10^1$
iPLD610/85C060	0.39 ns	$1.1 \times 10^{-1}$
iPLD910/85C090	0.28 ns	$1.5 \times 10^6$
iPLD22V10/85C22V10	0.43 ns	$3.24 \times 10^{10}$

Table 2. Maximum Clock Frequencies for Sampled PLDs

Margin = 0 ns FD = 4 MHz		Maximum Clock Frequency for a Chosen MTBF (Based on Equations 2 and 3)				
Device						
MTBF	5C032-30	5AC312-25	85C220-80	16V8A-10	16R6-7	85C090-15
1 Yr	16 MHz	22 MHz	53 MHz	46 MHz	26 MHz	34 MHz
10 Yrs	15 MHz	22 MHz	51 MHz	45 MHz	24 MHz	34 MHz
100 Yrs	15 MHz	21 MHz	50 MHz	44 MHz	23 MHz	33 MHz
1000 Yrs	15 MHz	21 MHz	49 MHz	43 MHz	21 MHz	32 MHz
FC = 33 MHz FD = 4 MHz		MTBFs for a Given T <sub>MET</sub> (Based on Equation 3)				
Device						
MTBF	5C032-30	85C220-80	16V8A-10	16R6-7	85C090-15	
1 Yr	22.1 ns	6.3 ns	3.9 ns	25.1 ns	8.9 ns	
10 Yrs	24.1 ns	6.8 ns	4.3 ns	27.7 ns	9.6 ns	
100 Yrs	26.1 ns	7.3 ns	4.6 ns	30.2 ns	10.2 ns	
1000 Yrs	28.0 ns	7.8 ns	5.0 ns	32.7 ns	10.8 ns	

## SUMMARY

Every dynamic system which has two stable states can enter a metastable state. Each PLD technology has a metastable characteristic associated with it, and some are less susceptible to metastability than others. By using Intel CMOS EPLDs and knowing how to effectively guard against the effects of metastability, systems may be designed to be virtually trouble-free.

## REFERENCES

1. Mark Johnson, MIPS Computer Systems, "Arbiter/Synchronizer Failure", COMP.ARCH, USENET Public Domain Computer Network, September 1989

2. Charles Dike, Signetics, *A Metastability Primer*, January 1989
3. Thomas Chaney, "Measured Flip Flop Responses To Marginal Triggering", *IEEE Transactions on Computers*, Volume C-32, No 12, December 1983
4. Peter Stoll, *How To Avoid Synchronization Problems*, VLSI Design, November/December 1982

## ACKNOWLEDGEMENTS

Thanks to members of Intel's engineering staff, especially to Mike Allen and Duane Chinnow for assistance with the conceptual background for this application note.

3. Thomas Chubb, "Measuring the Risk of Failure in  
Marginal Technology," IEEE Transactions on Com-  
puters, Volume C-32, No. 11, November 1983.  
4. Peter Stoll, How to Avoid System Reliability Problems,  
VLSI Design, November/December 1982.

#### ACKNOWLEDGEMENTS

Thanks to members of Intel's engineering staff espe-  
cially to Mike Allen and Duane Johnson for assistance  
with the conceptual background for this application  
note.

Every system designer which has two stable states can  
enter a metastable state. Each PLD technology has a  
metastable state associated with it and some  
designers are responsible to maintaining that state. By re-  
sisting the temptation to maintain a state and knowing how to effectively  
use the state, the effects of metastability can be avoided.  
It is not a design to be virtually trouble-free.

#### REFERENCES

1. Mark Johnson, "VLSI Computer Systems," (Addison-  
Wesley, Reading, MA), 1982.  
2. "Computer Architecture," (McGraw-Hill, New York), 1982.

November 1992

## PLD Quality and Reliability Data Summary

Order Number: 293003-004

# PLD QUALITY AND RELIABILITY DATA SUMMARY

CONTENTS	PAGE
THE IMPORTANCE OF RELIABILITY .....	3-236
EPD RELIABILITY DATA SUMMARY .....	3-236
REFERENCES .....	3-239

Intel's Reliability Monitor Program, devoted to assuring and controlling device reliability in production, is a proven tool that Intel has used for years and is now available to its customers. The Monitor Program supports all of Intel's technologies in a 40-year dynamic burn-in at 125°C (with a portion of these devices operating for a 1000-hour lifetime) and provides answers about device reliability that are not generally available from limited testing programs. Intel's much more than burn-in and device testing. When Intel releases the new devices, failure analysis is performed on each device part, including the bulk and determining the failure mechanism is a critical part of the Monitor Program. It is the most comprehensive reliability program any device manufacturer has.

The permanent objective is to deliver reliable, quality devices. Actions that Intel takes to meet this objective may include a process or design change or added reliability screen. Each decision is made with our customers in mind so that they receive the best and the performance that they expect in speed, cost, and reliability. This Reliability Monitor Program means that Intel's reliability standards are consistent with the highest standards in the industry. Our program ensures that Intel's reliability standards are consistently maintained day in and day out over the lifetime of a device's life. This reliability improves the life span of your product, reducing the required number of field service calls.

## EPD RELIABILITY DATA SUMMARY

Intel technology publishes this "EPD Reliability Data Summary," a continuing update of reliability information covering Intel's EPD product line. This document includes a discussion on EPD reliability testing technology and the most current failure rate calculations, failure analysis and field test results. The "EPD Reliability Data Summary" (order number 22047) and 22048 should be read as a supplement to monitor EPD product reliability.

Intel's commitment to the reliability of our products is clearly reflected in the information we make available to customers. We believe that supplying detailed reliability information to our customers is part of the total Intel effort and is an important part of Intel's leadership in microprocessor technology.

CONTENTS	PAGE
DEVICE DATA:	
85C22V10 .....	3-240
85C220 .....	3-245
85C224 .....	3-249
85C060 .....	3-253
85C090 .....	3-258
85C508 .....	3-263
5AC312 .....	3-267
5AC324 .....	3-271
5C032 .....	3-274
5C060 .....	3-278
5C090 .....	3-282
5C180 .....	3-286

APPENDIX A—FAILURE RATE CALCULATIONS FOR 60% UPPER CONFIDENCE LEVEL .....	3-289
---	-------

3

## The Roots of Reliability

The manufacture of a reliable semiconductor device is a dynamic and evolutionary process. Success of this process is highly dependent upon the synergy between knowledge and expertise. Intel's manufacturing engineers, materials physicists, and responsive management. Only the coordinated effort of these disciplines can consistently deliver high volumes of a reliable product. In this model, the experienced process engineers select and define the stress to be performed and the performance criteria to be met. The manufacturing engineers determine the test cases of the device and select failure modes and provide the solutions and recommendations. Finally, management provides the resources for the entire process from initial monitor to cost case control.



## THE IMPORTANCE OF RELIABILITY

Reliability of the erasable programmable logic devices (EPLDs) in your end product is critical to your total system reliability. The use of Intel EPLDs can make a difference. Intel EPLDs are manufactured on patented EPROM processes with proven reliability.

### Quality $\neq$ Reliability

A quality component is one that meets your specification when received and tested. A reliable component continues to meet your specification even years after you have shipped your product. While Intel is a quality leader, we also adhere to stringent reliability standards which we have established for ourselves.

### Consider Quality vs Reliability

The true cost of any component involves more than just the purchase price. The true component cost encompasses the initial purchase price, cost of rework during system production, and the cost of field repairs due to component failures. "Rework" costs during system production are incurred prior to shipment of your end product, and are a function of the quality of the component you purchase.

Repair costs incurred in the field after end product shipments, are a function of the reliability of the components. In addition to the increasing real cost of a system field service call, there is the intangible cost of a poor reliability reputation to the end user of your product. These costs depend upon the reliability of the components you purchase. Thus, reliability may impact costs during the system lifetime more than the initial quality of the components!

### The Roots of Reliability

The manufacture of a reliable semiconductor device using a modern technology is a dynamic and evolutionary process. Success of this process is highly dependent upon the interplay between knowledgeable and experienced manufacturing engineers, materials physicists, and responsible/responsive management. Only the correct combination can consistently deliver high volumes of a reliable product. In this model, the experienced process engineer selects and defines the stresses to be performed and the performance criteria to be met, utilizing appropriate statistical tools and limits. The materials physicist then determines the root cause of the failure, if and when failure occurs, and provides effective solutions and/or containment recommendations. Finally, management provides the resources for the entire process from initial monitor to root cause corrective action.

## Monitor Program

Reliability is designed into each component Intel manufactures. From the moment the design is put to paper, stringent reliability standards must be met at each step for a product to bear the Intel name.

Designing-in reliability, however, is only the beginning. Ongoing tests must be conducted to ensure that the original reliability specifications remain as valid in volume production as they were when the device was first qualified.

Intel's Reliability Monitor Program, devised to measure and control device reliability in production, is a proven tool that Intel has used for seven years and is now available to its customers. The Monitor Program subjects all of Intel's technologies to a 48 hour dynamic burn-in at 125°C (with a portion of these devices continued for a 1000 hour lifetest) and provides answers about device reliability that are not generally available from limited testing programs. But it's much more than burn-in and device testing. When test rejects are encountered, failure analysis is performed on each failed part. Isolating the fault and determining the failure mechanism is a critical part of the Monitor Program. It is the most comprehensive reliability program anywhere.

The paramount objective is to deliver reliable, quality devices. Actions that Intel takes to meet this objective may include a process or design change, or added reliability screen. Each decision is made with our customers in mind so that they receive the parts—and the performance—that they ordered by specifying Intel. Reliability qualification assures that all new production meets Intel's reliability standards. The Reliability Monitor Program ensures that these high standards are continually maintained, day in, day out, over the duration of a device's life. This reliability improves the life-time reputation of your product, reducing the required number of field service calls.

## EPLD RELIABILITY DATA SUMMARY

Intel routinely publishes this "EPLD Reliability Data Summary", a continuing update of reliability information covering Intel's EPLD product line. This document includes a discussion on EPLD reliability testing methodology and the most current failure rate calculations, failure analysis and lifetest results. The "EPROM Reliability Data Summary" (order number: 210473 and 293004) should be used as a supplement to monitor EPROM process reliability.

Intel's commitment to the reliability of our products is clearly reflected in the information we make available to customers. We believe that supplying detailed reliability information to our customers is part of the total solution Intel offers, and is an important part of Intel's leadership in microelectronics technology.

## EPLD Reliability Testing

Intel EPLDs undergo comprehensive testing to insure electrical reliability. This testing is done at qualification and/or during ongoing monitor checks. Testing differences between plastic and ceramic packaged EPLDs are noted.

Intel continually reviews its testing procedures and makes improvements to its methodology whenever overall reliability can be enhanced. Our goal is to be the industry leader in delivering high-quality, reliable parts.

Information on Intel's reliability testing procedures follows.

**High Temperature 5.25V Dynamic Lifetest**—This test is used to accelerate failure mechanisms by operating the devices at an elevated temperature of 125°C and nominal  $V_{CC}$ . During the test the device is programmed as a counter, with the inputs and outputs exercised, but not monitored or loaded. A pattern with greater than 90% of the EPROM cells programmed is used to simulate random customer patterns expected during actual use. Results of the lifetest have been summarized along with the failure analysis. Table 1 lists the activation energies for various failure mechanisms. These activation energies are used to calculate the amount of acceleration due to increased stress temperature or voltage (see Appendix A for failure rate calculations).

In order to best determine long-term failure rate, all devices used for lifetesting are first subjected to standard INTEL testing. The 48 hour burn-in results are an indication of infant mortality. These results are not included in the failure rate calculation. (See Figure 1 for typical lifetest bias and time diagrams.)

Table 1. Failure Mechanism Activation Energies

Failure Mechanism	$E_a$ (eV)
Oxide	0.3
Fab/Assembly Defect	0.5
SBCL/SBCG/MBCL/MBCG	0.6
Contamination	1.0
Speed Degradation	0.3–1.0
Intrinsic Charge Loss	1.4

## Failure Definitions

Oxide—An Oxide Failure Related Fault

SBCL—Single Bit Charge Loss

SBCG—Single Bit Charge Gain

MBCL—Multiple Bit Charge Loss

MBCG—Multiple Bit Charge Gain

Contamination—Ionic Contamination Failure

Speed Degradation—Device Speed Degraded Over Test

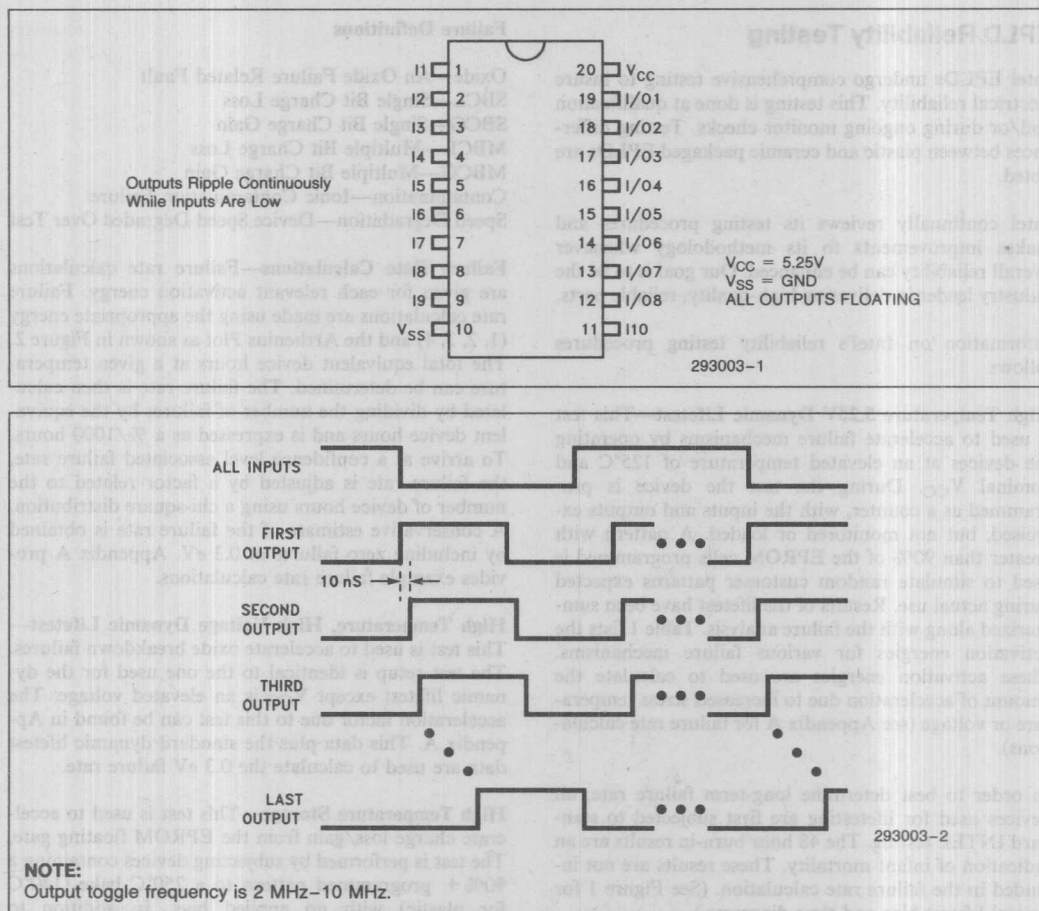
**Failure Rate Calculations**—Failure rate calculations are given for each relevant activation energy. Failure rate calculations are made using the appropriate energy (1, 2, 3, 4) and the Arrhenius Plot as shown in Figure 2. The total equivalent device hours at a given temperature can be determined. The failure rate is then calculated by dividing the number of failures by the equivalent device hours and is expressed as a %/1000 hours. To arrive at a confidence level associated failure rate, the failure rate is adjusted by a factor related to the number of device hours using a chi-square distribution. A conservative estimate of the failure rate is obtained by including zero failures at 0.3 eV. Appendix A provides example failure rate calculations.

## High Temperature, High Voltage Dynamic Lifetest

This test is used to accelerate oxide breakdown failures. The test setup is identical to the one used for the dynamic lifetest except  $V_{CC}$  is an elevated voltage. The acceleration factor due to this test can be found in Appendix A. This data plus the standard dynamic lifetest data are used to calculate the 0.3 eV failure rate.

**High Temperature Storage**—This test is used to accelerate charge loss/gain from the EPROM floating gate. The test is performed by subjecting devices containing a 90% + programmed pattern to a 250°C bake, (140°C for plastic) with no applied bias. In addition to EPROM cell integrity, this test is used to detect mechanical reliability problems (e.g., bond strength) and process stability. This test is sometimes referred to as Data Retention Bake Test.

**Temperature Cycle**—This test consists of cycling the temperature of the chamber housing the subject devices from  $-65^{\circ}\text{C}$  to  $+150^{\circ}\text{C}$  and back. The device is functionally tested before and after the stress. In addition, hermetic packages have fine/gross leak readouts. This test is to detect mechanical reliability problems and microcracks.



**Figure 1. EPDL Lifetest/Burn-In Bias and Timing Diagram  
(85C220 Used as Example)**

**85/85 Humidity**—High temperature/humidity testing is performed to evaluate moisture resistance characteristics of plastic-encapsulated components. A 2000-hour test is performed under static bias conditions at 85°C/85 percent relative humidity with nominal voltages. To maximum metal corrosion conditions, the biasing configuration is either under low power or no power, with alternate pins biased at +5V or 0V.

**Steam**—This test consists of exposing parts to water vapor at 121°C and at 2 atmospheres. The devices are functionally tested before and after the test. This test is used to highly accelerate failure mechanisms effecting the bonds, bond pads, and passivation.

**ESD Testing**—This test is performed to validate the products tolerance to Electro Static Discharge damage. All products incorporate ESD protection networks where needed to ensure the Intel corporate goals of military and charged device ESD testing are met. The military test uses the MIL STD 883 test criteria, while the charged device testing is performed to further validate protection occurring during mechanical handling.

**Programmability**—Device programmability is routinely monitored through the process of programming the devices for product monitors and qualifications. Programmability is a distinct part of a product qualification. All voltage combinations are verified. Program margin is measured and tested on ≥90% of Intel EPDL EPROM cells.

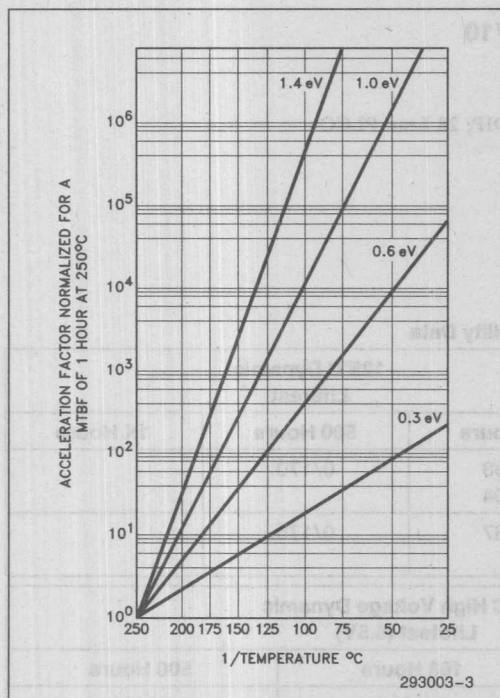


Figure 2. Arrhenius Plot

## REFERENCES

1. S. Rosenberg, D. Crook, B. Euzent, "16th Annual Proceedings of the International Reliability Physics Symposium," pp 19-25, 1978.
2. J. Caywood, B. Euzent, B. Shiner, "Data Retention in EPROMs," 1980 IEEE International Reliability Physics Symposium.
3. S. Rosenberg, B. Euzent, "HMOS Reliability" Reliability Report RR-18, Intel Corporation, 1979.
4. N. Mielke, "New EPROM Data-Loss Mechanisms," 1983. International Reliability Physics Symposium.
5. R. M. Alexander, "Calculating Failure Rates From Stress Data," April 1984 International Reliability Physics Symposium.
6. "EPROM Reliability Data Summary", Reliability Report RR-35, Intel Corporation, 1989, (updated regularly).
7. "EPROM Reliability Data Summary CHMOS IIIIE", Reliability Report RR-67, Intel Corporation, 1990, order number: 293004. (Updated Regularly).
8. "Intel Components Quality/Reliability Handbook", 1989, order number: 210997-004.
9. D. Baglee, L. Nanneman, C. Huang, "28th Annual Proceedings of the International Reliability Physics Symposium", pp. 12-18, 1990.
10. E. Anolick, G. Nelson, "17th Annual Proceedings of the International Reliability Physics Symposium", 1979.

## Reliability Data Summary

The following data is an accumulation of recent qualification and monitor program results. Failure rate calculation methods listed in Appendix A were used to arrive at the tabularized failure rates. Cerdip lifetest stresses are used to calculate the failure rate for each product and should also be used to indicate the failure rate of plastic products.

In reviewing the reliability data as presented, questions may arise as to why lot sizes often decrease from one test to another without a corresponding number of identified failures. This is due to a variety of factors. Many tests require smaller sample sizes and as a result all parts from a previous test do not necessarily flow through to a succeeding test.

In addition, various parts are pulled from a sample lot when mechanical or handler problems cause failures to occur. These "failures" are not a result of the specific test just completed. They are removed from the sample lot size and are not included in any failure rate calculation. It can also happen that a particular test is done incorrectly through human error or faulty test equipment and these suspected "invalid" failures are put aside for retesting at a later date, decreasing the lot size for a succeeding test. If these parts are found to be truly defective, they are treated as failures and listed. If they test out properly, they are removed from any calculation data base.



**85C22V10**

Device: 85C22V10  
 Organization: 10 Macrocells  
 Pinout: 24 Lead, 300 mil Cerdip and PDIP; 28 Lead PLCC  
 Die Size: 108 x 123 mils  
 Transistor Count: ~13K  
 Process: CHMOS IIIIE, P629.5  
 Programming Voltage: 12.5V  
 Technology: CMOS

**Table 1. Reliability Data**

Package	Burn-In 48 Hours	125°C Dynamic Lifetest		
		168 Hours	500 Hours	1K Hours
CERDIP	1/334	0/333	0/170	
PLCC	0/504	0/504		
TOTALS	1/838 A	0/837	0/170	

Package	125°C High Voltage Dynamic Lifetest (6.5V)		
	48 Hours	168 Hours	500 Hours
CERDIP	0/1819	0/1818	0/168
PDIP	0/336	0/336	
PLCC	0/814	0/814	
TOTALS	0/2969	0/2968	0/168

Package	Data Retention Bake (CERDIP @ 250°C, PDIP @ 140°C)		
	168 Hours	500 Hours	1K Hours
CERDIP	0/62		
PLCC	0/62		
TOTALS	0/124		

Table 2. Additional Qualification Stress Information

Package	Temperature Cycling (Condition "C"—150°C to -65°C)		
	200 Cycles	500 Cycles	1K Cycles
CERDIP	0/62		
PDIP	0/62		
PLCC	0/62		
TOTALS	0/186		

Package	85°C/85% Relative Humidity		
	168 Hours	500 Hours	1K Hours
PDIP	0/124		
PLCC	0/126		
TOTALS	0/250		

Package	Steam (121°C, 2 ATM)	
	96 Hours	168 Hours
PDIP	0/62	0/62
PLCC	0/62	0/62
TOTALS	0/124	0/124

3

Table 3. Failure Rate Predictions (CERDIP 85C22V10)

Actual Device Hrs	Ea (eV)	Equivalent Device Hrs		# Fail	Fail Rate in FITs (60% UCL)	
		55°C	70°C		55°C	70°C
9.54E + 4	0.3 LT	5.0E + 5	3.3E + 5			
2.66E + 5	0.3 HVLT * VAF	3.7E + 7	2.5E + 7			
Total 0.3 eV Failures =				0	0	0
9.54E + 4	0.5 LT	1.5E + 6	7.6E + 5			
2.66E + 5	0.5 HVLT	4.6E + 6	2.2E + 6			
Total 0.5 eV Failures =				0	158	312
9.54E + 4	0.6 LT	2.6E + 6	1.2E + 6			
2.66E + 5	0.6 HVLT	7.5E + 6	3.3E + 6			
Total 0.6 eV Failures =				0	0	0
9.54E + 4	1.0 LT	2.4E + 7	6.1E + 6			
2.66E + 5	1.0 HVLT	6.8E + 7	1.8E + 7			
Total 1.0 eV Failures =				0	0	0
Combined Failure Rate (FITs) =					158	312
48 Hour BI Infant Mortality = 1/2153 = 0.046%						

Table 4. Failure Rate Predictions (PDIP 85C22V10)

Actual Device Hrs	Ea (eV)	Equivalent Device Hrs		# Fail	Fail Rate in FITs (60% UCL)	
		55°C	70°C		55°C	70°C
6.0E + 4 1.4E + 5	0.3 LT 0.3 HVL * VAF	3.0E + 5 1.9E + 7	2.0E + 5 1.2E + 7			
Total 0.3 eV Failures =				0	0	0
6.0E + 4 1.4E + 5	0.5 LT 0.5 HVL	9.0E + 5 2.1E + 6	4.6E + 5 1.1E + 6			
Total 0.5 eV Failures =				0	306	595
6.0E + 4 1.4E + 5	0.6 LT 0.6 HVL	1.5E + 6 3.6E + 6	6.9E + 5 1.6E + 6			
Total 0.6 eV Failures =				0	0	0
6.0E + 4 1.4E + 5	1.0 LT 1.0 HVL	1.3E + 7 3.1E + 7	3.5E + 6 8.3E + 6			
Total 1.0 eV Failures =				0	0	0
Combined Failure Rate (FITs) =					306	595
48 Hour BI Infant Mortality = 0/1654 = 0.00%						

Table 5. Failure Rate Predictions (CERDIP and PDIP Combined Data 85C22V10)

Actual Device Hrs	Ea (eV)	Equivalent Device Hrs		# Fail	Fail Rate in FITs (60% UCL)	
		55°C	70°C		55°C	70°C
1.6E + 5 4.0E + 5	0.3 LT 0.3 HVL * VAF	8.1E + 5 5.3E + 7	5.4E + 5 3.5E + 7			
Total 0.3 eV Failures =				0	0	0
1.6E + 5 4.0E + 5	0.5 LT 0.5 HVL	2.4E + 6 6.0E + 6	1.2E + 6 3.1E + 6			
Total 0.5 eV Failures =				0	109	213
1.6E + 5 4.0E + 5	0.6 LT 0.6 HVL	4.1E + 6 1.0E + 7	1.9E + 6 4.6E + 6			
Total 0.6 eV Failures =				0	0	0
1.6E + 5 4.0E + 5	1.0 LT 1.0 HVL	3.6E + 7 9.0E + 7	9.5E + 6 2.4E + 7			
Total 1.0 eV Failures =				0	0	0
Combined Failure Rate (FITs) =					109	213
48 Hour BI Infant Mortality = 1/3807 = 0.026% = 532 DPM @ 60% Confidence						

# Other Data (CERDIP)

Theta Ja =	50°C/W	Temp with Theta Ja	
Theta Jc =	19°C/W	Degree Kelvin	
V <sub>CC</sub> =	5.25V	T(55°C) =	349.5 K
I <sub>CC</sub> @ 55°C =	75 mA	T(70°C) =	364.5 K
I <sub>CC</sub> @ 70°C =	75 mA	T(125°C) =	419.5 K
I <sub>CC</sub> @ 125°C =	75 mA	T(250°C) =	544.5 K

Boltzman's Constant =  $K = 8.62 \times 10^{-5} \text{ eV/K}$

# Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVL)			Bake (250°C)		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	5.26	3.49	0.3	n/a	n/a
0.5	15.9	8.04	0.5	730	337
0.6	27.7	12.2	0.6	2730	1080
1.0	253	64.7	1.0	n/a	n/a

# Other Data (PDIP)

Theta Ja =	66°C/W	Temp with Theta Ja	
Theta Jc =	21°C/W	Degree Kelvin	
V <sub>CC</sub> =	5.25V	T(55°C) =	354 K
I <sub>CC</sub> @ 55°C =	75 mA	T(70°C) =	369 K
I <sub>CC</sub> @ 70°C =	75 mA	T(125°C) =	424 K
I <sub>CC</sub> @ 125°C =	75 mA	T(250°C) =	549 K

Boltzman's Constant =  $K = 8.62 \times 10^{-5} \text{ eV/K}$

# Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVL)			Bake (140°C)		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	5.07	3.40	0.3	n/a	n/a
0.5	15.0	7.68	0.5	730	337
0.6	25.7	11.6	0.6	2730	1080
1.0	224	59.1	1.0	n/a	n/a



int

I <sub>neta</sub> Ja =	65°C/W
Theta Jc =	21°C/W
V <sub>CC</sub> =	5.25V
I <sub>CC</sub> @ 55°C =	75 mA
I <sub>CC</sub> @ 70°C =	75 mA
I <sub>CC</sub> @ 125°C =	75 mA

Temp with Theta Ja	
Degree Kelvin	
T(55°C) =	354 K
T(70°C) =	369 K
T(125°C) =	424 K
T(250°C) =	549 K

Boltzman's Constant =  $K = 8.62 \times 10^{-5} \text{ eV/K}$

## Thermal Acceleration Factors—

### Lifetest (LT) and High Voltage Lifetest (HVLTL)

Activation Energy	55°C	70°C
0.3	5.07	3.40
0.5	15.0	7.68
0.6	25.7	11.6
1.0	224	59.1

### Bake (140°C)

Activation Energy	55°C	70°C
0.3	n/a	n/a
0.5	730	337
0.6	2730	1080
1.0	n/a	n/a

## Failure Analysis

A. ISB failure, destroyed in analysis.

# 85C220

Device: 85C220  
 Organization: 8 Macrocells  
 Pinout: 24 Lead, 300 mil Cerdip and PDIP; 20 Lead PLCC  
 Die Size: 89 x 90 mils  
 Transistor Count: ~7K  
 Process: CHMOS IIIIE, P629.5  
 Programming Voltage: 12.5V  
 Technology: CMOS

Table 1. Reliability Data

Package	Burn-In 48 Hours	125°C Dynamic Lifetest		
		168 Hours	500 Hours	1K Hours
CERDIP	0/2050	0/2050	0/270	0/270
PDIP	0/1995	0/1995	0/485	0/270
TOTALS	0/4045	0/4045	0/755	0/540
Additional Readouts:				
CERDIP	0/90 @ 2K Hours Lifetest			
PDIP	0/180 @ 2K Hours Lifetest			

Package	125°C High Voltage Dynamic Lifetest (6.5V)			
	48 Hours	168 Hours	500 Hours	1K Hours
CERDIP	0/2048	0/2048	0/270	0/90
PDIP	0/400	0/399	0/299	0/199
TOTALS	0/2448	0/2447	0/569	0/289
Additional Readouts:				
CERDIP	0/90 @ 2K Hours Lifetest			
PDIP	0/199 @ 2K Hours Lifetest			

Package	Data Retention Bake (CERDIP @ 250°C, PDIP @ 140°C)		
	168 Hours	500 Hours	1K Hours
CERDIP	0/300	0/300	
PDIP	0/300	0/300	
TOTALS	0/600	0/600	

Table 2. Additional Qualification Stress Information

Package	Temperature Cycling (Condition "C"—150°C to -65°C)		
	500 Cycles	1K Cycles	2K Cycles
CERDIP	0/90	0/89	0/89
PDIP	0/150	0/150	0/150
PLCC	0/78	0/78	
TOTALS	0/318	0/317	0/239

Package	85°C/85% Relative Humidity		
	500 Hours	1K Hours	2K Hours
PDIP	0/98	0/98	0/48
PLCC	0/30	0/30	
TOTALS	0/128	0/128	0/48

Package	Steam (121°C, 2 ATM)		
	168 Hours	336 Hours	500 Hours
PDIP	0/300	0/300	0/300
PLCC	0/100	0/100	
TOTALS	0/400	0/400	0/300

Table 3. Failure Rate Predictions (CERDIP 85C220)

Actual Device Hrs	Ea (eV)	Equivalent Device Hrs		# Fail	Fail Rate in FITs (60% UCL)	
		55°C	70°C		55°C	70°C
5.6E + 5	0.3 LT	3.2E + 6	2.1E + 6			
5.6E + 5	0.3 HVLT * VAF	8.3E + 7	5.4E + 7			
Total 0.3 eV Failures =				0	0	0
5.6E + 5	0.5 LT	1.0E + 7	4.98E + 6			
5.6E + 5	0.5 HVLT	1.0E + 7	4.98E + 6			
Total 0.5 eV Failures =				0	44	91
5.6E + 5	0.6 LT	1.8E + 7	7.7E + 6			
5.6E + 5	0.6 HVLT	1.8E + 7	7.7E + 6			
Total 0.6 eV Failures =				0	0	0
5.6E + 5	1.0 LT	1.9E + 8	4.4E + 7			
5.6E + 5	1.0 HVLT	1.9E + 8	4.4E + 7			
Total 1.0 eV Failures =				0	0	0
Combined Failure Rate (FITs) =					44	91
48 Hour BI Infant Mortality = 0/4098 = 0.00%						

Table 4. Failure Rate Predictions (PDIP 85C220)

Actual Device Hrs	Ea (eV)	Equivalent Device Hrs		# Fail	Fail Rate in FITs (60% UCL)	
		55°C	70°C		55°C	70°C
7.20E + 5 5.0E + 5	0.3 LT 0.3 HVL * VAF	4.0E + 6 7.2E + 7	2.6E + 6 4.7E + 7			
Total 0.3 eV Failures =				0	0	0
7.20E + 5 5.0E + 5	0.5 LT 0.5 HVL	1.2E + 7 8.6E + 6	6.1E + 6 4.25E + 6			
Total 0.5 eV Failures =				0	44	88
7.20E + 5 5.0E + 5	0.6 LT 0.6 HVL	2.2E + 7 1.5E + 7	9.4E + 6 6.5E + 6			
Total 0.6 eV Failures =				0	0	0
7.20E + 5 5.0E + 5	1.0 LT 1.0 HVL	2.1E + 8 1.5E + 8	5.2E + 7 3.6E + 7			
Total 1.0 eV Failures =				0	0	0
Combined Failure Rate (FITs) =					44	88
48 Hour BI Infant Mortality = 0/2395 = 0.00%						

3

Table 5. Failure Rate Predictions (CERDIP and PDIP Combined Data)

Actual Device Hrs	Ea (eV)	Equivalent Device Hrs		# Fail	Fail Rate in FITs (60% UCL)	
		55°C	70°C		55°C	70°C
1.3E + 6 1.1E + 6	0.3 LT 0.3 HVL * VAF	7.0E + 6 1.5E + 8	4.6E + 6 1.0E + 8			
Total 0.3 eV Failures =				0	0	0
1.3E + 6 1.1E + 6	0.5 LT 0.5 HVL	2.2E + 7 1.8E + 7	1.1E + 7 9.0E + 6			
Total 0.5 eV Failures =				0	23	46
1.3E + 6 1.1E + 6	0.6 LT 0.6 HVL	3.9E + 7 3.2E + 7	1.7E + 7 1.4E + 7			
Total 0.6 eV Failures =				0	0	0
1.3E + 6 1.1E + 6	1.0 LT 1.0 HVL	3.7E + 8 3.1E + 8	9.2E + 7 7.7E + 7			
Total 1.0 eV Failures =				0	0	0
Combined Failure Rate (FITs) =					23	46
48 Hour BI Infant Mortality = 0/2225 = 0.00% = 142 DPM @ 60% Confidence						



## Other Data (CERDIP)

Theta Ja =	68°C/W	Temp with Theta Ja	
Theta Jc =	30°C/W	Degree Kelvin	
V <sub>CC</sub> =	5.25V	T(55°C) =	340.5 K
I <sub>CC</sub> @ 55°C =	35 mA	T(70°C) =	355.5 K
I <sub>CC</sub> @ 70°C =	35 mA	T(125°C) =	410.5 K
I <sub>CC</sub> @ 125°C =	35 mA	T(250°C) =	535.5 K
Boltzman's Constant = K = $8.62 \times 10^{-5}$ eV/K			

## Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLt)			Bake		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	5.71	3.71	0.3	n/a	n/a
0.5	18.3	8.90	0.5	730	337
0.6	32.7	13.8	0.6	2730	1080
1.0	334	79.2	1.0	n/a	n/a

## Other Data (PDIP and PLCC)

Theta Ja =	90°C/W	Temp with Theta Ja	
Theta Jc =	25°C/W	Degree Kelvin	
V <sub>CC</sub> =	5.25V	T(55°C) =	344.5 K
I <sub>CC</sub> @ 55°C =	35 mA	T(70°C) =	359.5 K
I <sub>CC</sub> @ 70°C =	35 mA	T(125°C) =	414.5 K
I <sub>CC</sub> @ 125°C =	35 mA	T(250°C) =	539.5 K
Boltzman's Constant = K = $8.62 \times 10^{-5}$ eV/K			

## Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLt)			Bake		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	5.51	3.61	0.3	n/a	n/a
0.5	17.2	8.50	0.5	730	337
0.6	30.3	13.1	0.6	2730	1080
1.0	295	72.3	1.0	n/a	n/a

## Failure Analysis

N/A

Device: 85C224  
 Organization: 8 Macrocells  
 Pinout: 24 Lead, 300 mil CERPDP and PDIP; 28 Lead PLCC  
 Die Size: 99 x 94 mils  
 Transistor Count: ~ 7K  
 Process: CHMOS IIIIE, P629.5  
 Programming Voltage: 12.5V  
 Technology: CMOS

Table 1. Reliability Data

Package	Burn-In 48 Hours	125°C Dynamic Lifetest		
		168 Hours	500 Hours	1K Hours
CERPDP	0/897	0/897	0/255	0/255
PDIP	0/300	0/300	0/170	0/170
TOTALS	0/1197	0/1197	0/425	0/425

Package	125°C High Voltage Dynamic Lifetest (6.5V)			
	48 Hours	168 Hours	500 Hours	1K Hours
CERPDP	0/728	1/450	0/449	0/449
PDIP	0/300	0/297	0/297	0/297
TOTALS	0/1028	1/747 (A)	0/746	0/746

Package	Data Retention Bake (CERPDP @ 250°C, PDIP @ 140°C)		
	168 Hours	500 Hours	1K Hours
CERPDP	0/300	0/300	0/300
PDIP	0/200	0/200	
TOTALS	0/500	0/500	0/300

Table 2. Additional Qualification Stress Information

Package	Temperature Cycling (Condition "C"—150°C to -65°C)		
	500 Cycles	1K Cycles	2K Cycles
CERPDP	0/90	0/90	
PDIP	0/60	0/60	
PLCC	0/60	0/60	
TOTALS	0/210	0/210	

Package	Steam (121°C, 2 ATM)		
	168 Hours	336 Hours	500 Hours
PDIP	0/200	0/200	
PLCC	0/100	0/100	
TOTALS	0/300	0/300	

Actual Device Hrs	Ea (eV)	Equivalent Device Hrs		# Fail	Fail Rate in FITs (60% UCL)	
		55°C	70°C		55°C	70°C
3.2E + 5 4.3E + 5	0.3 LT 0.3 HVL * VAF	1.9E + 6 6.5E + 7	1.2E + 6 4.2E + 7			
Total 0.3 eV Failures =				0	30	46
3.2E + 5 4.3E + 5	0.5 LT 0.5 HVL	6.1E + 6 8.2E + 6	2.9E + 6 3.9E + 6			
Total 0.5 eV Failures =				1	142	294
3.2E + 5 4.3E + 5	0.6 LT 0.6 HVL	1.1E + 7 1.5E + 7	4.6E + 6 6.1E + 6			
Total 0.6 eV Failures =				0	0	0
3.2E + 5 4.3E + 5	1.0 LT 1.0 HVL	1.2E + 8 1.6E + 8	2.7E + 7 3.6E + 7			
Total 1.0 eV Failures =				0	0	0
Combined Failure Rate (FITs) =					172	330
48 Hour BI Infant Mortality = 0/1625 = 0.00%						

**Table 4. Failure Rate Predictions (PDIP 85C224)**

Actual Device Hrs	Ea (eV)	Equivalent Device Hrs		# Fail	Fail Rate in FITs (60% UCL)	
		55°C	70°C		55°C	70°C
1.8E + 5 2.8E + 5	0.3 LT 0.3 HVL * VAF	1.0E + 6 4.1E + 7	6.6E + 5 2.7E + 7			
Total 0.3 eV Failures =				0	0	0
1.8E + 5 2.8E + 5	0.5 LT 0.5 HVL	3.2E + 6 5.0E + 6	1.6E + 6 2.5E + 6			
Total 0.5 eV Failures =				0	111	227
1.8E + 5 2.8E + 5	0.6 LT 0.6 HVL	5.7E + 6 8.9E + 6	2.4E + 6 3.8E + 6			
Total 0.6 eV Failures =				0	0	0
1.8E + 5 2.8E + 5	1.0 LT 1.0 HVL	5.8E + 7 9.0E + 7	1.4E + 7 2.2E + 7			
Total 1.0 eV Failures =				0	0	0
Combined Failure Rate (FITs) =					111	227
48 Hour BI Infant Mortality = 0/600 = 0.00%						

Table 5. Failure Rate Predictions (CERDIP and PDIP Combined Data)

Actual Device Hrs	Ea (eV)	Equivalent Device Hrs		# Fail	Fail Rate in FITs (60% UCL)	
		55°C	70°C		55°C	70°C
5.0E + 5 7.1E + 5	0.3 LT 0.3 HVLT * VAF	2.8E + 6 1.0E + 8	1.8E + 6 6.8E + 7			
Total 0.3 eV Failures =				0	19	29
5.0E + 5 7.1E + 5	0.5 LT 0.5 HVLT	9.0E + 6 1.3E + 7	4.4E + 6 6.2E + 6			
Total 0.5 eV Failures =				1	93	190
5.0E + 5 7.1E + 5	0.6 LT 0.6 HVLT	1.6E + 7 2.3E + 7	6.8E + 6 9.6E + 6			
Total 0.6 eV Failures =				0	0	0
5.0E + 5 7.1E + 5	1.0 LT 1.0 HVLT	1.6E + 8 2.3E + 8	3.9E + 7 5.5E + 7			
Total 1.0 eV Failures =				0	0	0
Combined Failure Rate (FITs) =					112	219
48 Hour BI Infant Mortality = $0/2225 = 0.00\% = 412 \text{ DPM @ 60\% Confidence}$						

3

## Other Data (CERDIP)

Theta Ja =	55°C/W	Temp with Theta Ja	
Theta Jc =	20°C/W	Degree Kelvin	
V <sub>CC</sub> =	5.25V	T(55°C) =	338.1 K
I <sub>CC</sub> @ 55°C =	35 mA	T(70°C) =	353.1 K
I <sub>CC</sub> @ 70°C =	35 mA	T(125°C) =	408.1 K
I <sub>CC</sub> @ 125°C =	35 mA	T(250°C) =	533.1 K

Boltzman's Constant =  $K = 8.62 \times 10^{-5} \text{ eV/K}$ 

## Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLT)			Bake		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	5.84	3.77	0.3	n/a	n/a
0.5	19.0	9.15	0.5	730	337
0.6	34.2	14.2	0.6	2730	1080
1.0	360	83.7	1.0	n/a	n/a



**Other Data (PDIP)**

Theta Ja =	75°C/W	<b>Temp with Theta Ja</b>	
Theta Jc =	22°C/W	<b>Degree Kelvin</b>	
V <sub>CC</sub> =	5.25V	T(55°C) =	341.8 K
I <sub>CC</sub> @ 55°C =	35 mA	T(70°C) =	356.8 K
I <sub>CC</sub> @ 70°C =	35 mA	T(125°C) =	411.8 K
I <sub>CC</sub> @ 125°C =	35 mA	T(250°C) =	536.8 K

Boltzman's Constant =  $K = 8.62 \times 10^{-5} \text{ eV/K}$

**Thermal Acceleration Factors—**

<b>Lifetest (LT) and High Voltage Lifetest (HVL)</b>			<b>Bake</b>		
<b>Activation Energy</b>	<b>55°C</b>	<b>70°C</b>	<b>Activation Energy</b>	<b>55°C</b>	<b>70°C</b>
0.3	5.65	3.68	0.3	n/a	n/a
0.5	17.9	8.77	0.5	730	337
0.6	31.9	13.5	0.6	2730	1080
1.0	321	76.9	1.0	n/a	n/a

**Other Data (PLCC)**

Theta Ja =	75°C/W	<b>Temp with Theta Ja</b>	
Theta Jc =	22°C/W	<b>Degree Kelvin</b>	
V <sub>CC</sub> =	5.25V	T(55°C) =	341.8 K
I <sub>CC</sub> @ 55°C =	35 mA	T(70°C) =	356.8 K
I <sub>CC</sub> @ 70°C =	35 mA	T(125°C) =	411.8 K
I <sub>CC</sub> @ 125°C =	35 mA	T(250°C) =	536.8 K

Boltzman's Constant =  $K = 8.62 \times 10^{-5} \text{ eV/K}$

**Thermal Acceleration Factors—**

<b>Lifetest (LT) and High Voltage Lifetest (HVL)</b>			<b>Bake</b>		
<b>Activation Energy</b>	<b>55°C</b>	<b>70°C</b>	<b>Activation Energy</b>	<b>55°C</b>	<b>70°C</b>
0.3	5.65	3.68	0.3	n/a	n/a
0.5	17.9	8.77	0.5	730	337
0.6	31.9	13.5	0.6	2730	1080
1.0	321	76.9	1.0	n/a	n/a

**Failure Analysis**

A. Leakage failure, 0.5 eV

**Intel**

Organization: 16 B Macrocells  
 Pinout: 24 Lead, 300 mil Cerdip and PDIP; 28 Lead PLCC  
 Die Size: 112 x 136 mils  
 Transistor Count: ~14K  
 Process: CHMOS IIIIE, P629.5  
 Programming Voltage: 12.5V  
 Technology: CMOS

**Table 1. Reliability Data**

Package	Burn-In 48 Hours	125°C Dynamic Lifetest		
		168 Hours	500 Hours	1K Hours
CERDIP	1/900	0/899	0/270	0/270
PDIP	0/600	0/600	0/180	0/180
TOTALS	1/1500 A	0/1499	0/450	0/450

Package	125°C High Voltage Dynamic Lifetest (6.5V)		
	48 Hours	168 Hours	500 Hours
CERDIP	0/450	0/450	0/450
PDIP	0/300	0/300	0/300
TOTALS	0/750	0/750	0/750

Additional Readouts:

CERDIP 0/450 @ 1K Hours HVLT  
 PDIP 0/450 @ 1K Hours HVLT; 0/450 @ 2K Hours HVLT

Package	Data Retention Bake (CERDIP @ 250°C, PDIP @ 140°C)		
	168 Hours	500 Hours	1K Hours
CERDIP	0/300	0/300	
PDIP	0/200	0/200	
TOTALS	0/500	0/500	

**3**

Table 2. Additional Qualification Stress Information

Package	Temperature Cycling (Condition "C"—150°C to -65°C)		
	500 Cycles	1K Cycles	2K Cycles
CERDIP	0/90	0/90	
PDIP	0/60	0/60	
PLCC	0/110	0/110	
TOTALS	0/260	0/260	

Package	Steam (121°C, 2 ATM)	
	96 Hours	168 Hours
PDIP	0/200	0/200
PLCC	0/100	1/100
TOTALS	0/300	1/300

Table 3. Failure Rate Predictions (CERDIP 85C060)

Actual Device Hrs	Ea (eV)	Equivalent Device Hrs		# Fail	Fail Rate in FITs (60% UCL)	
		55°C	70°C		55°C	70°C
3.3E + 5	0.3 LT	1.8E + 6	1.2E + 6			
4.3E + 5	0.3 HVL * VAF	6.0E + 7	4.0E + 7			
Total 0.3 eV Failures =				0	0	0
3.3E + 5	0.5 LT	5.5E + 6	2.8E + 6			
4.3E + 5	0.5 HVL	7.1E + 6	3.6E + 6			
Total 0.5 eV Failures =				0	72	145
3.3E + 5	0.6 LT	9.7E + 6	4.2E + 6			
4.3E + 5	0.6 HVL	1.3E + 7	5.4E + 6			
Total 0.6 eV Failures =				0	0	0
3.3E + 5	1.0 LT	9.2E + 7	2.3E + 7			
4.3E + 5	1.0 HVL	1.2E + 8	3.0E + 7			
Total 1.0 eV Failures =				0	0	0
Combined Failure Rate (FITs) =					72	145
48 Hour BI Infant Mortality = 1/2250 = 0.044%						

Table 4. Failure Rate Predictions (PDIP 85C060)

Actual Device Hrs	Ea (eV)	Equivalent Device Hrs		# Fail	Fail Rate in FITs (60% UCL)	
		55°C	70°C		55°C	70°C
2.2E + 5 2.9E + 5	0.3 LT 0.3 HVL * VAF	1.2E + 6 3.9E + 7	7.8E + 5 2.6E + 7			
Total 0.3 eV Failures =				0	0	0
2.2E + 5 2.9E + 5	0.5 LT 0.5 HVL	3.6E + 6 4.6E + 6	1.8E + 6 2.3E + 6			
Total 0.5 eV Failures =				0	112	223
2.2E + 5 2.9E + 5	0.6 LT 0.6 HVL	6.2E + 6 8.0E + 6	2.7E + 6 3.5E + 6			
Total 0.6 eV Failures =				0	0	0
2.2E + 5 2.9E + 5	1.0 LT 1.0 HVL	5.7E + 7 7.4E + 7	1.5E + 7 1.9E + 7			
Total 1.0 eV Failures =				0	0	0
Combined Failure Rate (FITs) =					112	223
48 Hour BI Infant Mortality = $0/900 = 0.00\%$						

3

Table 5. Failure Rate Predictions (CERDIP and PDIP Combined Data 85C060)

Actual Device Hrs	Ea (eV)	Equivalent Device Hrs		# Fail	Fail Rate in FITs (60% UCL)	
		55°C	70°C		55°C	70°C
5.6E + 5 7.1E + 5	0.3 LT 0.3 HVL * VAF	2.9E + 6 9.8E + 7	2.0E + 6 6.5E + 7			
Total 0.3 eV Failures =				0	0	0
5.6E + 5 7.1E + 5	0.5 LT 0.5 HVL	8.9E + 6 1.2E + 7	4.5E + 6 5.8E + 6			
Total 0.5 eV Failures =				0	45	89
5.6E + 5 7.1E + 5	0.6 LT 0.6 HVL	1.6E + 7 2.0E + 7	6.8E + 6 8.8E + 6			
Total 0.6 eV Failures =				0	0	0
5.6E + 5 7.1E + 5	1.0 LT 1.0 HVL	1.4E + 8 1.8E + 8	3.6E + 7 4.7E + 7			
Total 1.0 eV Failures =				0	0	0
Combined Failure Rate (FITs) =					45	89
48 Hour BI Infant Mortality = $1/2250 = 0.04\% = 899 \text{ DPM @ } 60\% \text{ Confidence}$						



## Other Data (CERDIP)

Theta Ja =	59°C/W	Temp with Theta Ja Degree Kelvin			
Theta Jc =	18°C/W				
V <sub>CC</sub> =	5.25V	T(55°C) =	346.6 K		
I <sub>CC</sub> @ 55°C =	60 mA	T(70°C) =	361.6 K		
I <sub>CC</sub> @ 70°C =	60 mA	T(125°C) =	416.6 K		
I <sub>CC</sub> @ 125°C =	60 mA	T(250°C) =	541.6 K		
Boltzman's Constant = K = $8.62 \times 10^{-5}$ eV/K					

## Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLT)			Bake		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	5.41	3.56	0.3	n/a	n/a
0.5	16.6	8.31	0.5	730	337
0.6	29.2	12.7	0.6	2730	1080
1.0	277	69.1	1.0	n/a	n/a

## Other Data (PDIP)

Theta Ja =	67°C/W	Temp with Theta Ja Degree Kelvin			
Theta Jc =	45°C/W				
V <sub>CC</sub> =	5.25V	T(55°C) =	349.1 K		
I <sub>CC</sub> @ 55°C =	60 mA	T(70°C) =	364.1 K		
I <sub>CC</sub> @ 70°C =	60 mA	T(125°C) =	419.1 K		
I <sub>CC</sub> @ 125°C =	60 mA	T(250°C) =	544.1 K		
Boltzman's Constant = K = $8.62 \times 10^{-5}$ eV/K					

## Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLT)			Bake		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	5.29	3.51	0.3	n/a	n/a
0.5	16.0	8.09	0.5	730	337
0.6	27.9	12.3	0.6	2730	1080
1.0	257	65.4	1.0	n/a	n/a

Theta Ja = 65°C/W      Temp with Theta Ja  
 Theta Jc = 16°C/W      Degree Kelvin  
 V<sub>CC</sub> = 5.25V      T(55°C) = 349.1 K  
 I<sub>CC</sub> @ 55°C = 60 mA      T(70°C) = 364.1 K  
 I<sub>CC</sub> @ 70°C = 60 mA      T(125°C) = 419.1 K  
 I<sub>CC</sub> @ 125°C = 60 mA      T(250°C) = 544.1 K  
 Boltzman's Constant = K =  $8.62 \times 10^{-5}$  eV/K

### Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLT)			Bake		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	5.29	3.51	0.3	n/a	n/a
0.5	16.0	8.09	0.5	730	337
0.6	27.9	12.3	0.6	2730	1080
1.0	257	65.4	1.0	n/a	n/a

### Failure Analysis

A. Charge Loss due to Passivation Damage.

Data Retention Bake (CERDIP @ 250°C, PDIP @ 140°C)			Additional Factors:		
1K Hours	500 Hours	100 Hours	0.125 @ 1K-Hours HVLT	0.125 @ 1K-Hours HVLT	Package
	0.1200	0.1200			CERDIP
	0.1200	0.1200			PDIP
	0.1200	0.1200			TOTALS

Device: 85C090  
 Organization: 24 Macrocells  
 Pinout: 40 Lead, 600 mil Cerdip and PDIP; 44 Lead PLCC  
 Die Size: 136 x 152 mils  
 Transistor Count: ~ 30K  
 Process: CHMOS IIIE, P629.5  
 Programming Voltage: 12.5V  
 Technology: CMOS

**Table 1. Reliability Data**

Package	Burn-In 48 Hours	125°C Dynamic Lifetest		
		168 Hours	500 Hours	1K Hours
CERDIP	0/900	0/900	0/90	0/90
PDIP	0/600	0/600		
TOTALS	0/1500	0/1500	0/90	0/90

Package	125°C High Voltage Dynamic Lifetest (6.5V)		
	48 Hours	168 Hours	500 Hours
CERDIP	0/450	0/450	0/270
PDIP	0/300	0/300	0/240
TOTALS	0/750	0/750	0/510

Additional Readouts:			
CERDIP	0/270 @ 1K Hours HVLT		
PDIP	0/240 @ 1K Hours HVLT		

Package	Data Retention Bake (CERDIP @ 250°C, PDIP @ 140°C)		
	168 Hours	500 Hours	1K Hours
CERDIP	0/300	0/300	
PDIP	0/200	0/200	
TOTALS	0/500	0/500	

Table 2. Additional Qualification Stress Information

Package	Temperature Cycling (Condition "C"—150°C to -65°C)		
	500 Cycles	1K Cycles	2K Cycles
CERDIP	0/90	0/90	
PDIP	0/60	0/60	
PLCC	0/149	0/149	
TOTALS	0/299	0/299	

Package	Steam (121°C, 2 ATM)	
	96 Hours	168 Hours
PDIP	0/200	0/200
PLCC	0/100	0/100
TOTALS	0/300	0/300

Table 3. Failure Rate Predictions (CERDIP 85C090)

Actual Device Hrs	Ea (eV)	Equivalent Device Hrs		# Fail	Fail Rate in FITs (60% UCL)	
		55°C	70°C		55°C	70°C
9.0E + 4	0.3 LT	4.6E + 5	3.1E + 5			
2.7E + 5	0.3 HVL * VAF	3.6E + 7	2.4E + 7			
Total 0.3 eV Failures =				0	0	0
9.0E + 4	0.5 LT	1.4E + 6	7.0E + 5			
2.7E + 5	0.5 HVL	4.1E + 6	2.1E + 6			
Total 0.5 eV Failures =				0	166	326
9.0E + 4	0.6 LT	2.4E + 6	1.1E + 6			
2.7E + 5	0.6 HVL	7.1E + 6	3.2E + 6			
Total 0.6 eV Failures =				0	0	0
9.0E + 4	1.0 LT	2.1E + 7	5.5E + 6			
2.7E + 5	1.0 HVL	6.3E + 7	1.6E + 7			
Total 1.0 eV Failures =				0	0	0
Combined Failure Rate (FITs) =					166	326
48 Hour BI Infant Mortality = 0/1350 = 0.00%						



Table 4. Failure Rate Predictions (PDIP 85C090)

Actual Device Hrs	Ea (eV)	Equivalent Device Hrs		# Fail	Fail Rate in FITs (60% UCL)	
		55°C	70°C		55°C	70°C
7.2E + 4 2.4E + 5	0.3 LT 0.3 HVL * VAF	3.6E + 5 3.1E + 7	2.4E + 5 2.1E + 7			
Total 0.3 eV Failures =				0	0	0
7.2E + 4 2.4E + 5	0.5 LT 0.5 HVL	1.1E + 6 3.5E + 6	5.4E + 5 1.8E + 6			
Total 0.5 eV Failures =				0	202	390
7.2E + 4 2.4E + 5	0.6 LT 0.6 HVL	1.8E + 6 6.0E + 6	8.1E + 5 2.7E + 6			
Total 0.6 eV Failures =				0	0	0
7.2E + 4 2.4E + 5	1.0 LT 1.0 HVL	1.5E + 7 5.1E + 7	4.1E + 6 1.4E + 7			
Total 1.0 eV Failures =				0	0	0
Combined Failure Rate (FITs) =					202	390
48 Hour BI Infant Mortality = 0/900 = 0.00%						

Table 5. Failure Rate Predictions (CERDIP and PDIP Combined Data 85C090)

Actual Device Hrs	Ea (eV)	Equivalent Device Hrs		# Fail	Fail Rate in FITs (60% UCL)	
		55°C	70°C		55°C	70°C
1.6E + 5 5.1E + 5	0.3 LT 0.3 HVL * VAF	8.0E + 5 6.6E + 7	5.4E + 5 4.5E + 7			
Total 0.3 eV Failures =				0	0	0
1.6E + 5 5.1E + 5	0.5 LT 0.5 HVL	2.3E + 6 7.4E + 6	1.2E + 6 3.8E + 6			
Total 0.5 eV Failures =				0	94	182
1.6E + 5 5.1E + 5	0.6 LT 0.6 HVL	4.0E + 6 1.3E + 7	1.8E + 6 5.7E + 6			
Total 0.6 eV Failures =				0	0	0
1.6E + 5 5.1E + 5	1.0 LT 1.0 HVL	3.4E + 7 1.1E + 8	9.1E + 6 2.9E + 7			
Total 1.0 eV Failures =				0	0	0
Combined Failure Rate (FITs) =					94	182
48 Hour BI Infant Mortality = 0/2250 = 0.00% = 408 DPM @ 60% Confidence						

**Other Data (CERDIP)**

Theta Ja =	44.5°C/W	<b>Temp with Theta Ja</b>	
Theta Jc =	17°C/W	<b>Degree Kelvin</b>	
V <sub>CC</sub> =	5.25V	T(55°C) =	353.5 K
I <sub>CC</sub> @ 55°C =	105 mA	T(70°C) =	368.5 K
I <sub>CC</sub> @ 70°C =	105 mA	T(125°C) =	423.5 K
I <sub>CC</sub> @ 125°C =	105 mA	T(250°C) =	548.5 K
Boltzman's Constant = $K = 8.62 \times 10^{-5} \text{ eV/K}$			

**Thermal Acceleration Factors—**

Lifetest (LT) and High Voltage Lifetest (HVLT)			Bake (250°C)		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	5.13	3.43	0.3	n/a	n/a
0.5	15.3	7.8	0.5	730	337
0.6	26.3	11.8	0.6	2730	1080
1.0	233	60.9	1.0	n/a	n/a

**Other Data (PDIP)**

Theta Ja =	51°C/W	<b>Temp with Theta Ja</b>	
Theta Jc =	29°C/W	<b>Degree Kelvin</b>	
V <sub>CC</sub> =	5.25V	T(55°C) =	356.1 K
I <sub>CC</sub> @ 55°C =	105 mA	T(70°C) =	371.1 K
I <sub>CC</sub> @ 70°C =	105 mA	T(125°C) =	426.1 K
I <sub>CC</sub> @ 125°C =	105 mA	T(250°C) =	551.1 K
Boltzman's Constant = $K = 8.62 \times 10^{-5} \text{ eV/K}$			

**Thermal Acceleration Factors—**

Lifetest (LT) and High Voltage Lifetest (HVLT)			Bake (140°C)		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	4.98	3.35	0.3	n/a	n/a
0.5	14.5	7.52	0.5	730	337
0.6	24.8	11.3	0.6	2730	1080
1.0	211	56.5	1.0	n/a	n/a

## Other Data (PLCC)

Theta Ja =	55°C/W	Temp with Theta Ja	
Theta Jc =	16°C/W	Degree Kelvin	
V <sub>CC</sub> =	5.25V	T(55°C) =	356.1 K
I <sub>CC</sub> @ 55°C =	105 mA	T(70°C) =	371.1 K
I <sub>CC</sub> @ 70°C =	105 mA	T(125°C) =	426.1 K
I <sub>CC</sub> @ 125°C =	105 mA	T(250°C) =	551.1 K

Boltzman's Constant =  $K = 8.62 \times 10^{-5} \text{ eV/K}$ 

## Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLT)			Bake (140°C)		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	4.98	3.35	0.3	n/a	n/a
0.5	14.5	7.52	0.5	730	337
0.6	24.8	11.3	0.6	2730	1080
1.0	211	56.5	1.0	n/a	n/a

## Failure Analysis

N/A

## 85C508

Device: 85C508  
 Organization: Decoder/Latch PLD  
 Pinout: 28 Lead, 300 mil CERDIP and PDIP; 28 Lead PLCC  
 Die Size: 65 x 98 mils  
 Transistor Count: ~10K  
 Process: CHMOS IIIIE, P629.5  
 Programming Voltage: 12.5V  
 Technology: CMOS

Table 1. Reliability Data

Package	Burn-In 48 Hours	125°C Dynamic Lifetest		
		168 Hours	1K Hours	2K Hours
CERDIP	0/1200	0/1195	0/270	
PDIP	1/800	0/799	0/180	
TOTALS	1/2000 A	0/1994	0/450	

Package	125°C High Voltage Dynamic Lifetest (6.5V)		
	48 Hours	168 Hours	1K Hours
CERDIP	1/599	0/599	0/270
PDIP	0/399	0/399	0/180
TOTALS	1/998	0/998	0/450

Package	Data Retention Bake (CERDIP @ 250°C, PDIP @ 140°C)		
	168 Hours	500 Hours	1K Hours
CERDIP	0/300	0/300	
PDIP	0/200	0/200	
TOTALS	0/500	0/500	



Table 2. Additional Qualification Stress Information

Package	Temperature Cycling (Condition "C"—150°C to -65°C)		
	500 Cycles	1K Cycles	2K Cycles
CERDIP	0/150	0/150	
PDIP	0/100	0/100	
PLCC	0/60	0/60	
TOTALS	0/310	0/310	

Package	Steam (121°C, 2 ATM)		
	168 Hours	336 Hours	500 Hours
PDIP	0/200	0/200	0/200
PLCC	1/600	0/60	
TOTALS	0/260	0/260	0/200

Table 3. Failure Rate Predictions (CERDIP 85C508)

Actual Device Hrs	Ea (eV)	Equivalent Device Hrs		# Fail	Fail Rate in FITs (60% UCL)	
		55°C	70°C		55°C	70°C
3.7E + 5	0.3 LT	2.2E + 6	1.4E + 6			
3.0E + 5	0.3 HVL * VAF	4.5E + 7	2.9E + 7			
Total 0.3 eV Failures =				0	0	0
3.7E + 5	0.5 LT	7.1E + 6	3.4E + 6			
3.0E + 5	0.5 HVL	5.7E + 6	2.7E + 6			
Total 0.5 eV Failures =				0	71	151
3.7E + 5	0.6 LT	1.3E + 7	5.2E + 6			
3.0E + 5	0.6 HVL	1.0E + 7	4.2E + 6			
Total 0.6 eV Failures =				0	0	0
3.7E + 5	1.0 LT	1.4E + 8	3.1E + 7			
3.0E + 5	1.0 HVL	1.1E + 8	2.5E + 7			
Total 1.0 eV Failures =				0	0	0
Combined Failure Rate (FITs) =					71	151
48 Hour BI Infant Mortality = 0/1799 = 0.00%						

Table 4. Failure Rate Predictions (PDIP 85C508)

Actual Device Hrs	Ea (eV)	Equivalent Device Hrs		# Fail	Fail Rate in FITs (60% UCL)	
		55°C	70°C		55°C	70°C
2.5E + 5 2.0E + 5	0.3 LT 0.3 HVL * VAF	1.4E + 6 3.0E + 7	9.1E + 5 1.9E + 7			
Total 0.3 eV Failures =				0	0	0
2.5E + 5 2.0E + 5	0.5 LT 0.5 HVL	4.6E + 6 3.7E + 6	2.2E + 6 1.8E + 6			
Total 0.5 eV Failures =				0	111	232
2.5E + 5 2.0E + 5	0.6 LT 0.6 HVL	8.2E + 6 6.6E + 6	3.4E + 6 2.7E + 6			
Total 0.6 eV Failures =				0	0	0
2.5E + 5 2.0E + 5	1.0 LT 1.0 HVL	8.5E + 7 6.8E + 7	1.9E + 7 1.6E + 6			
Total 1.0 eV Failures =				0	0	0
Combined Failure Rate (FITs) =					111	232
48 Hour BI Infant Mortality = 1/1199 = 0.08%						

3

Table 5. Failure Rate Predictions (CERDIP and PDIP Combined Data)

Actual Device Hrs	Ea (eV)	Equivalent Device Hrs		# Fail	Fail Rate in FITs (60% UCL)	
		55°C	70°C		55°C	70°C
6.1E + 5 5.0E + 5	0.3 LT 0.3 HVL * VAF	3.6E + 6 7.4E + 7	2.3E + 6 4.8E + 7			
Total 0.3 eV Failures =				0	0	0
6.1E + 5 5.0E + 5	0.5 LT 0.5 HVL	1.1E + 7 9.2E + 6	5.5E + 6 4.4E + 6			
Total 0.5 eV Failures =				0	44	92
6.1E + 5 5.0E + 5	0.6 LT 0.6 HVL	2.1E + 7 1.7E + 7	8.5E + 6 6.8E + 6			
Total 0.6 eV Failures =				0	0	0
6.1E + 5 5.0E + 5	1.0 LT 1.0 HVL	2.1E + 8 1.7E + 8	4.9E + 7 3.9E + 7			
Total 1.0 eV Failures =				0	0	0
Combined Failure Rate (FITs) =					44	92
48 Hour BI Infant Mortality = 1/2998 = 0.03% = 675 DPM @ 60% Confidence						

Theta Ja =	83°C/W	<b>Temp with Theta Ja</b>	
Theta Jc =	18°C/W	<b>Degree Kelvin</b>	
V <sub>CC</sub> =	5.25V	T(55°C) =	335 K
I <sub>CC</sub> @ 55°C =	15 mA	T(70°C) =	350 K
I <sub>CC</sub> @ 70°C =	15 mA	T(125°C) =	403 K
I <sub>CC</sub> @ 125°C =	12 mA	T(250°C) =	523 K

Boltzman's Constant =  $K = 8.62 \times 10^{-5} \text{ eV/K}$

### Thermal Acceleration Factors—

<b>Lifetest (LT) and High Voltage Lifetest (HVL)</b>			<b>Bake (250°C)</b>		
<b>Activation Energy</b>	<b>55°C</b>	<b>70°C</b>	<b>Activation Energy</b>	<b>55°C</b>	<b>70°C</b>
0.3	5.88	3.76	0.3	n/a	n/a
0.5	19.2	9.10	0.5	727	336
0.6	34.6	14.2	0.6	2718	1075
1.0	367	82.8	1.0	n/a	n/a

### Other Data (PLCC)

Theta Ja =	100°C/W	<b>Temp with Theta Ja</b>	
Theta Jc =	23°C/W	<b>Degree Kelvin</b>	
V <sub>CC</sub> =	5.25V	T(55°C) =	336 K
I <sub>CC</sub> @ 55°C =	15 mA	T(70°C) =	350 K
I <sub>CC</sub> @ 70°C =	15 mA	T(125°C) =	403 K
I <sub>CC</sub> @ 125°C =	12 mA	T(250°C) =	413 K

Boltzman's Constant =  $K = 8.62 \times 10^{-5} \text{ eV/K}$

### Thermal Acceleration Factors—

<b>Lifetest (LT) and High Voltage Lifetest (HVL)</b>			<b>Bake (140°C)</b>		
<b>Activation Energy</b>	<b>55°C</b>	<b>70°C</b>	<b>Activation Energy</b>	<b>55°C</b>	<b>70°C</b>
0.3	5.77	3.71	0.3	n/a	n/a
0.5	18.6	8.87	0.5	38.0	17.5
0.6	33.3	13.7	0.6	78.7	31.1
1.0	345	78.8	1.0	n/a	n/a

### Failure Analysis

A. Isb Failure — 0.3 eV

## 5AC312

Device: 5AC312  
 Organization: 12 Macrocells  
 Pinout: 24 Lead, 300 mil Cerdip and PDIP (28 Lead PLCC)  
 Die Size: 130 x 189 mils  
 Transistor Count: ~ 17K  
 Process: CHMOS IIIE, P429  
 Programming Voltage: 12.5V  
 Technology: CMOS

Table 1. Reliability Data

Package	Burn-In 48 Hours	125°C Dynamic Lifetest		
		168 Hours	500 Hours	1K Hours
CERDIP	1/3960	0/2955	0/183	0/183
TOTALS	1/3960 A	0/2955	0/183	0/183
Additional Readouts: CERDIP 0/183 @ 2K Hours Lifetest				
Package	125°C High Voltage Dynamic Lifetest (7.0V)			
	48 Hours	168 Hours	500 Hours	
CERDIP	0/97	0/93	0/92	
TOTALS	0/97	0/93	0/92	
Additional Readouts: CERDIP 0/80 @ 1K Hours High Voltage Lifetest				
Package	Data Retention Bake (CERDIP @ 250°C)			
	48 Hours	168 Hours	500 Hours	
CERDIP	0/100	0/99	0/99	
TOTALS	0/100	0/99	0/99	
Additional Readouts: CERDIP 0/96 @ 1K Hours, 250°C Bake				



Table 2. Additional Qualification Stress Information

Package	Temperature Cycling (Condition "C"—150°C to -65°C)		
	200 Cycles	500 Cycles	1K Cycles
PDIP	0/234	0/234	0/234
TOTALS	0/234	0/234	0/234
Package	85°C/85% Relative Humidity		
	168 Hours	500 Hours	1K Hours
PDIP	0/219	0/219	0/219
TOTALS	0/219	0/219	0/219
Additional Readouts: PDIP 0/219 @ 2K Hours			
Package	Thermal Shock (Condition "C"—150°C to -65°C)		
	50 Cycles	200 Cycles	500 Cycles
PDIP	0/231	0/231	0/231
TOTALS	0/231	0/231	0/231
Package	Steam (121°C, 2 ATM)		
	96 Hours	168 Hours	
PDIP	0/393	0/393	
PLCC	0/304	0/304	
TOTALS	0/697	0/697	
Additional Readouts: PDIP 0/234 @ 336 Hours of Steam			

Table 3. Failure Rate Predictions (5AC312-CERDIP-Fab 1)

Actual Device Hrs	Ea (eV)	Equivalent Device Hrs		# Fail	Fail Rate in FITs (60% UCL)	
		55°C	70°C		55°C	70°C
6.90E + 05	0.3 LT	3.42E + 06	2.26E + 06	0		
8.64E + 04	0.3 HVLT * VAF	3.98E + 07	2.63E + 07	0		
Total 0.3 eV Failures =				0	0.0	0.0
6.90E + 05	0.5 LT	9.96E + 06	5.00E + 06	0		
8.64E + 04	0.5 HVLT	1.25E + 06	6.26E + 05	0		
Total 0.5 eV Failures =				0	81.7	162.7
6.90E + 05	0.6 LT	1.70E + 07	7.42E + 06	0		
8.64E + 04	0.6 HVLT	2.13E + 06	9.29E + 05	0		
Total 0.6 eV Failures =				0	0.0	0.0
6.90E + 05	1.0 LT	1.44E + 08	3.62E + 07	0		
8.64E + 04	1.0 HVLT	1.80E + 07	4.53E + 06	0		
Total 1.0 eV Failures =				0	0.0	0.0
Combined Failure Rate (FITs) =					81.7	162.7
48 Hour BI Infant Mortality = $1/3960 = 0.0253\% = 511 \text{ DPM @ } 60\% \text{ Confidence}$						

3

## Other Data (CERDIP)

Theta Ja =	47°C/W	Temp with Theta Ja	
Theta Jc =	16°C/W	Degree Kelvin	
V <sub>CC</sub> =	5.25V	T(55°C) =	348 K
I <sub>CC</sub> @ 55°C =	80 mA	T(70°C) =	363 K
I <sub>CC</sub> @ 70°C =	80 mA	T(125°C) =	414 K
I <sub>CC</sub> @ 125°C =	65 mA	T(250°C) =	523 K

Boltzman's Constant =  $K = 8.62 \times 10^{-5} \text{ eV/K}$ 

## Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLT)			Bake (250°C)		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	4.96	3.28	0.3	n/a	n/a
0.5	14.43	7.24	0.5	727.61	335.91
0.6	24.61	10.76	0.6	2718.18	1075.15
1.0	208.28	52.47	1.0	n/a	n/a

# Other Data (PLCC)

Theta Ja =	52°C/W	Temp with Theta Ja Degree Kelvin		
Theta Jc =	16°C/W			
V <sub>CC</sub> =	5.25V	T(55°C) =	349.84 K	
I <sub>CC</sub> @ 55°C =	80 mA	T(70°C) =	364.84 K	
I <sub>CC</sub> @ 70°C =	80 mA	T(125°C) =	415.75 K	
I <sub>CC</sub> @ 125°C =	65 mA	T(140°C) =	413 K	

Boltzman's Constant = K =  $8.62 \times 10^{-5}$  eV/K

## Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLT)			Bake (140°C)		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	4.84	3.21	0.3	n/a	n/a
0.5	13.83	7.00	0.5	38.00	17.54
0.6	23.39	10.33	0.6	78.65	31.11
1.0	191.27	48.96	1.0	n/a	n/a

## Other Data (PLCC)

Theta Ja =	56°C/W	Temp with Theta Ja Degree Kelvin		
Theta Jc =	16°C/W			
V <sub>CC</sub> =	5.25V	T(55°C) =	351.52 K	
I <sub>CC</sub> @ 55°C =	80 mA	T(70°C) =	366.52 K	
I <sub>CC</sub> @ 70°C =	80 mA	T(125°C) =	417.11 K	
I <sub>CC</sub> @ 125°C =	65 mA	T(140°C) =	413 K	

Boltzman's Constant = K =  $8.62 \times 10^{-5}$  eV/K

## Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLT)			Bake (140°C)		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	4.74	3.16	0.3	n/a	n/a
0.5	13.37	6.81	0.5	38.00	17.54
0.6	22.46	9.99	0.6	78.65	31.11
1.0	178.85	46.37	1.0	n/a	n/a

Voltage Acceleration Factor (VAF) for HVLT on this process is 26 (6.5V) and 93 (7.0V).

### NOTE:

FIT = Failure in Time. 1 FIT = 1 failure per  $1 \times 10^9$  device hours.

## Failure Analysis

A. 1 oxide failure — 0.3 eV

**5AC324**

Device: 5AC324  
 Organization: 24 Macrocells  
 Pinout: 40 Lead, 600 mil CERDIP and PDIP (44 Lead PLCC)  
 Die Size: 241 x 187 mils  
 Transistor Count: ~68K  
 Process: CHMOS IIIE, P429  
 Programming Voltage: 12.5V  
 Technology: CMOS

**Table 1. Reliability Data**

Package	Burn-In 48 Hours	125°C Dynamic Lifetest		
		168 Hours	500 Hours	1K Hours
CERDIP	4/5721	0/2972	0/401	0/401
TOTALS	4/5721 A	0/2972	0/401	0/401
Additional Readouts: CERDIP 0/103 @ 2K Hours Lifetest				
Package	125°C High Voltage Dynamic Lifetest (7.0V)			
	48 Hours	168 Hours	500 Hours	
CERDIP	0/113	0/113	0/113	
TOTALS	0/113	0/113	0/113	
Additional Readouts: CERDIP 0/113 @ 1K Hours High Voltage Lifetest				
Package	Data Retention Bake (CERDIP @ 250°C)			
	48 Hours	168 Hours	500 Hours	
CERDIP	0/99	0/99	0/99	
TOTALS	0/99	0/99	0/99	
Additional Readouts: CERDIP 0/99 @ 1K Hours, 250°C Bake				



Table 2. Additional Qualification Stress Information

Package	Temperature Cycling (Condition "C"—150°C to -65°C)		
	200 Cycles	500 Cycles	1000 Cycles
CERDIP	0/176	0/176	0/176
PDIP	0/48	0/48	0/48
Additional Readouts:			
CERDIP	0/176 @ 1700 Cycles		
PDIP	0/48 @ 1700 Cycles		
Package	85°C/85% Relative Humidity		
	168 Hours	500 Hours	1K Hours
PDIP	0/291	0/291	0/291
TOTALS	0/291	0/291	0/291
Package	Steam (121°C, 2 ATM)		
	96 Hours	168 Hours	
PDIP	0/104	0/104	
PLCC	0/481	0/481	
TOTALS	0/585	0/585	
Additional Readouts:			
PDIP	0/104 @ 500 Hours of Steam		
PLCC	0/333 @ 500 Hours of Steam		

Table 3. Failure Rate Predictions (5AC324-CERDIP-Fab 1)

Actual Device Hrs	Ea (eV)	Equivalent Device Hrs		# Fail	Fail Rate in FITs (60% UCL)	
		55°C	70°C		55°C	70°C
7.93E + 05	0.3 LT	3.66E + 06	2.46E + 06	0		
1.13E + 05	0.3 HVL * VAF	4.85E + 07	3.26E + 07	0		
Total 0.3 eV Failures =				0	0.0	0.0
7.93E + 05	0.5 LT	1.01E + 07	5.22E + 06	0		
1.13E + 05	0.5 HVL	1.44E + 06	7.44E + 05	0		
Total 0.5 eV Failures =				0	79.0	153.3
7.93E + 05	0.6 LT	1.69E + 07	7.62E + 06	0		
1.13E + 05	0.6 HVL	2.40E + 06	1.09E + 06	0		
Total 0.6 eV Failures =				0	0.0	0.0
7.93E + 05	1.0 LT	1.30E + 08	3.44E + 07	0		
1.13E + 05	1.0 HVL	1.85E + 07	4.90E + 06	0		
Total 1.0 eV Failures =				0	0.0	0.0
Combined Failure Rate (FITs) =					79.0	153.3
48 Hour BI Infant Mortality = 4/5721 = 0.0699% = 916 DPM @ 60% Confidence						

## Other Data (CERDIP)

Theta Ja =	34°C/W	<b>Temp with Theta Ja</b>	
Theta Jc =	13°C/W	<b>Degree Kelvin</b>	
V <sub>CC</sub> =	5.25V	T(55°C) =	355 K
I <sub>CC</sub> @ 55°C =	150 mA	T(70°C) =	370 K
I <sub>CC</sub> @ 70°C =	150 mA	T(125°C) =	420 K
I <sub>CC</sub> @ 125°C =	125 mA	T(250°C) =	523 K
Boltzman's Constant = K = $8.62 \times 10^{-5}$ eV/K			

## Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLТ)			Bake (250°C)		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	4.61	3.10	0.3	n/a	n/a
0.5	12.78	6.59	0.5	727.61	335.91
0.6	21.27	9.60	0.6	2718.18	1075.15
1.0	163.32	43.38	1.0	n/a	n/a

## Other Data (PDIP and PLCC)

Theta Ja =	43°C/W	<b>Temp with Theta Ja</b>	
Theta Jc =	15°C/W	<b>Degree Kelvin</b>	
V <sub>CC</sub> =	5.25V	T(55°C) =	361.86 K
I <sub>CC</sub> @ 55°C =	150 mA	T(70°C) =	376.86 K
I <sub>CC</sub> @ 70°C =	150 mA	T(125°C) =	426.22 K
I <sub>CC</sub> @ 125°C =	125 mA	T(140°C) =	413 K
Boltzman's Constant = K = $8.62 \times 10^{-5}$ eV/K			

## Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLТ)			Bake (140°C)		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	4.27	2.91	0.3	n/a	n/a
0.5	11.23	5.94	0.5	38.00	17.54
0.6	18.22	8.48	0.6	78.65	31.11
1.0	126.20	35.25	1.0	n/a	n/a

Voltage Acceleration Factor (VAF) for HVLТ on this process is 26 (6.5V) and 93 (7.0V).

## NOTE:

FIT = Failure in Time. 1 FIT = 1 failure per  $1 \times 10^9$  device hours.

## Failure Analysis

- A. 2 passivation crack — 0.5 eV  
 1 metal stringer — 0.5 eV  
 1 oxide defect — 0.3 eV

# 5C032

Device: 5C032  
 Organization: 8 Macrocells  
 Pinout: 20 Lead, 300 mil CERDIP and PDIP  
 Die Size: 86 x 93 mils  
 Transistor Count: ~ 7K  
 Process: CHMOS IIE, P424  
 Programming Voltage: 12.5V  
 Technology: CMOS

Table 1. Reliability Data

Package	Burn-In 48 Hours	125°C Dynamic Lifetest		
		168 Hours	500 Hours	1K Hours
CERDIP	0/2956	0/2956	0/299	2/283
PDIP	1/2661	0/2660	0/149	0/146
TOTALS	1/5617 A	0/5616	0/448	2/429 B
Additional Readouts: CERDIP 0/281 @ 2K Hours Lifetest				
Package	125°C High Voltage Dynamic Lifetest (6.5V)			
	48 Hours	168 Hours	500 Hours	
CERDIP	0/200	0/200	0/179	
PDIP	0/150	1/150	0/149	
TOTALS	0/350	1/350 C	0/328	
Additional Readouts: CERDIP 0/129 @ 1K Hours High Voltage Lifetest PDIP 0/146 @ 1K Hours High Voltage Lifetest				
Package	Data Retention Bake (CERDIP @ 250°C, PDIP @ 140°C)			
	48 Hours	168 Hours	500 Hours	
CERDIP	2/277	0/146	0/146	
PDIP	0/300	0/298	0/298	
TOTALS	2/577 D	0/444	0/444	
Additional Readouts: PDIP 0/298 @ 1K Hours, 140°C Bake				

Table 2. Additional Qualification Stress Information

Package	Temperature Cycling (Condition "C"—150°C to -65°C)		
	200 Cycles	500 Cycles	1K Cycles
CERDIP	0/232	0/75	0/75
PDIP	0/150	0/149	0/149
TOTALS	0/382	0/224	0/224
Package	Thermal Shock (Condition "C"—150°C to -65°C)		
	50 Cycles	200 Cycles	500 Cycles
PDIP	0/150	0/150	0/150
TOTALS	0/150	0/150	0/150
Package	85°C/85% Relative Humidity		
	168 Hours	500 Hours	1K Hours
PDIP	0/263	0/263	1/263
TOTALS	0/263	0/263	1/263 E
Package	Steam (121°C, 2 ATM)		
	96 Hours	168 Hours	
PDIP	0/262	0/262	
Additional Readouts: PDIP 0/262 @ 500 Hours of Steam			



Table 3. Failure Rate Predictions (CERDIP)

Actual Device Hrs	Ea (eV)	Equivalent Device Hrs		# Fail	Fail Rate in FITs (60% UCL)	
		55°C	70°C		55°C	70°C
8.76E + 05	0.3 LT	4.76E + 06	3.10E + 06	0		
1.58E + 05	0.3 HVL * VAF	4.70E + 07	3.06E + 07	0		
Total 0.3 eV Failures =				0	0.0	0.0
8.76E + 05	0.5 LT	1.47E + 07	7.18E + 06	0		
1.58E + 05	0.5 HVL	2.64E + 06	1.29E + 06	0		
Total 0.5 eV Failures =				0	52.8	108.0
8.76E + 05	0.6 LT	2.58E + 07	1.09E + 07	0		
1.58E + 05	0.6 HVL	4.64E + 06	1.97E + 06	0		
Total 0.6 eV Failures =				0	0.0	0.0
8.76E + 05	1.0 LT	2.46E + 08	5.88E + 07	2		
1.58E + 05	1.0 HVL	4.43E + 07	1.06E + 07	0		
Total 1.0 eV Failures =				2	10.7	44.7
Combined Failure Rate (FITs) =					63.5	152.8
48 Hour BI Infant Mortality = $0/2956 = 0.0000\% = 310 \text{ DPM @ 60\% Confidence}$						

## Other Data (CERDIP)

Theta Ja =	83°C/W	Temp with Theta Ja	
Theta Jc =	20°C/W	Degree Kelvin	
V <sub>CC</sub> =	5.25V	T(55°C) =	341.07 K
I <sub>CC</sub> @ 55°C =	30 mA	T(70°C) =	356.07 K
I <sub>CC</sub> @ 70°C =	30 mA	T(125°C) =	408.89 K
I <sub>CC</sub> @ 125°C =	25 mA	T(250°C) =	523 K
Boltzman's Constant = $K = 8.62 \times 10^{-5} \text{ eV/K}$			

## Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVL)			Bake (250°C)		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	5.43	3.53	0.3	n/a	n/a
0.5	16.76	8.19	0.5	727.61	335.91
0.6	29.46	12.48	0.6	2718.18	1075.15
1.0	280.99	67.11	1.0	n/a	n/a

Table 4. Failure Rate Predictions (PDIP)

Actual Device Hrs	Ea (eV)	Equivalent Device Hrs		# Fail	Fail Rate in FITs (60% UCL)	
		55°C	70°C		55°C	70°C
4.42E + 05 1.48E + 05	0.3 LT 0.3 HVLT * VAF	2.28E + 06 4.19E + 07	1.50E + 06 2.75E + 07	0 0		
Total 0.3 eV Failures =				0	20.7	31.5
4.42E + 05 1.48E + 05	0.5 LT 0.5 HVLT	6.81E + 06 2.28E + 06	3.38E + 06 1.13E + 06	0 1		
Total 0.5 eV Failures =				1	222.5	447.7
4.42E + 05 1.48E + 05	0.6 LT 0.6 HVLT	1.18E + 07 3.93E + 06	5.08E + 06 1.70E + 06	0 0		
Total 0.6 eV Failures =				0	0.0	0.0
4.42E + 05 1.48E + 05	1.0 LT 1.0 HVLT	1.05E + 08 3.51E + 07	2.59E + 07 8.65E + 06	0 0		
Total 1.0 eV Failures =				0	0.0	0.0
Combined Failure Rate (FITs) =					243.2	479.3
48 Hour BI Infant Mortality = $1/2661 = 0.0376\% = 760 \text{ DPM @ } 60\% \text{ Confidence}$						

3

## Other Data (PDIP)

Theta Ja =	109°C/W	Temp with Theta Ja	
Theta Jc =	20°C/W	Degree Kelvin	
V <sub>CC</sub> =	5.25V	T(55°C) =	345.17 K
I <sub>CC</sub> @ 55°C =	30 mA	T(70°C) =	360.17 K
I <sub>CC</sub> @ 70°C =	30 mA	T(125°C) =	412.31 K
I <sub>CC</sub> @ 125°C =	25 mA	T(140°C) =	413 K
Boltzman's Constant = $K = 8.62 \times 10^{-5} \text{ eV/K}$			

## Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLT)			Bake (140°C)		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	5.16	3.39	0.3	n/a	n/a
0.5	15.41	7.66	0.5	38.00	17.54
0.6	26.62	11.50	0.6	78.65	31.11
1.0	237.39	58.60	1.0	n/a	n/a

Voltage Acceleration Factor (VAF) for HVLT on this process is 55.

## NOTE:

FIT = Failure in Time. 1 FIT = 1 failure per  $1 \times 10^9$  device hours.

## Failure Analysis

- A. 1 leakage — 0.5 eV
- B. 2 leakage — 1.0 eV
- C. 1 leakage — 0.5 eV

- D. 1 SBCL — 0.6 eV
- 1 passivation defect — 0.5 eV
- E. 1 passivation defect — 0.5 eV

## 5C060

Device: 5C060  
 Organization: 16 Macrocells  
 Pinout: 24 Lead, 300 mil Cerdip and PDIP (28 Lead PLCC)  
 Die Size: 135 x 141 mils  
 Transistor Count: ~ 14K  
 Process: CHMOS IIE, P424  
 Programming Voltage: 12.5V  
 Technology: CMOS

Table 1. Reliability Data

Package	Burn-In 48 Hours	125°C Dynamic Lifetest		
		168 Hours	500 Hours	1K Hours
CERDIP	0/3969 1/2500	0/3969 3/2499	0/371 0/100	0/354 0/99
PDIP		0/351	0/351	0/351
TOTALS	1/6469 A	3/6819 B	0/882	0/804
Additional Readouts:				
CERDIP		0/99 @ 2K Hours Lifetest		
PDIP		0/351 @ 2K Hours Lifetest		
Package	125°C High Voltage Dynamic Lifetest (6.5V)			
	48 Hours	168 Hours	500 Hours	
CERDIP	0/224 0/100	0/201 0/100	2/198 0/100	
TOTALS	0/324	0/301	2/298 C	
Package	Data Retention Bake (CERDIP @ 250°C)			
	48 Hours	168 Hours	500 Hours	
CERDIP	0/203 0/50	0/202 0/50	0/202 0/50	
TOTALS	0/253	0/252	0/252	
Additional Readouts:				
CERDIP		0/127 @ 1K Hours, 250°C Bake		

Table 2. Additional Qualification Stress Information

Package	Temperature Cycling (Condition "C"—150°C to -65°C)		
	200 Cycles	500 Cycles	1K Cycles
CERDIP	0/307	1/307	0/302
PDIP	0/350	0/350	—
TOTALS	0/657	1/657 D	0/302

Package	Thermal Shock (Condition "C"—150°C to -65°C)		
	50 Cycles	200 Cycles	500 Cycles
CERDIP	0/155	0/155	1/155
TOTALS	0/155	0/155	1/155 E

Additional Readouts:  
CERDIP 0/154 @ 1K Cycles

Package	Steam (121°C, 2 ATM)	
	96 Hours	168 Hours
PDIP	0/100	0/100
PLCC	0/150	0/150
TOTALS	0/250	0/250

Additional Readouts:  
PDIP 0/100 @ 336 Hours of Steam

Table 3. Failure Rate Predictions (5C060-CERDIP-Fab 1)

Actual Device Hrs	Ea (eV)	Equivalent Device Hrs		# Fail	Fail Rate in FITs (60% UCL)	
		55°C	70°C		55°C	70°C
1.26E + 06	0.3 LT	6.37E + 06	4.24E + 06	2		
1.51E + 05	0.3 HVL * VAF	4.20E + 07	2.79E + 07	2		
Total 0.3 eV Failures =				4	108.6	163.2
1.26E + 06	0.5 LT	1.88E + 07	9.54E + 06	1		
1.51E + 05	0.5 HVL	2.25E + 06	1.14E + 06	0		
Total 0.5 eV Failures =				1	96.0	189.1
1.26E + 06	0.6 LT	3.23E + 07	1.43E + 07	0		
1.51E + 05	0.6 HVL	3.86E + 06	1.71E + 06	0		
Total 0.6 eV Failures =				0	0.0	0.0
1.26E + 06	1.0 LT	2.81E + 08	7.24E + 07	0		
1.51E + 05	1.0 HVL	3.36E + 07	8.66E + 06	0		
Total 1.0 eV Failures =				0	0.0	0.0
Combined Failure Rate (FITs) =					204.6	352.2
48 Hour BI Infant Mortality = 1/6469 = 0.0155% = 313 DPM @ 60% Confidence						



Theta Ja =	54°C/W	<b>Temp with Theta Ja</b>	
Theta Jc =	17°C/W	<b>Degree Kelvin</b>	
V <sub>CC</sub> =	5.25V	T(55°C) =	350.68 K
I <sub>CC</sub> @ 55°C =	80 mA	T(70°C) =	365.68 K
I <sub>CC</sub> @ 70°C =	80 mA	T(125°C) =	419.26 K
I <sub>CC</sub> @ 125°C =	75 mA	T(250°C) =	523 K

Boltzman's Constant =  $K = 8.62 \times 10^{-5} \text{ eV/K}$

### Thermal Acceleration Factors—

<b>Lifetest (LT) and High Voltage Lifetest (HVLt)</b>			<b>Bake (250°C)</b>		
<b>Activation Energy</b>	<b>55°C</b>	<b>70°C</b>	<b>Activation Energy</b>	<b>55°C</b>	<b>70°C</b>
0.3	5.07	3.37	0.3	n/a	n/a
0.5	14.94	7.58	0.5	727.61	335.91
0.6	25.66	11.37	0.6	2718.18	1075.15
1.0	223.24	57.52	1.0	n/a	n/a

### Other Data (PDIP)

Theta Ja =	67°C/W	<b>Temp with Theta Ja</b>	
Theta Jc =	22°C/W	<b>Degree Kelvin</b>	
V <sub>CC</sub> =	5.25V	T(55°C) =	356.14 K
I <sub>CC</sub> @ 55°C =	80 mA	T(70°C) =	371.14 K
I <sub>CC</sub> @ 70°C =	80 mA	T(125°C) =	424.38 K
I <sub>CC</sub> @ 125°C =	75 mA	T(140°C) =	413 K

Boltzman's Constant =  $K = 8.62 \times 10^{-5} \text{ eV/K}$

### Thermal Acceleration Factors—

<b>Lifetest (LT) and High Voltage Lifetest (HVLt)</b>			<b>Bake (140°C)</b>		
<b>Activation Energy</b>	<b>55°C</b>	<b>70°C</b>	<b>Activation Energy</b>	<b>55°C</b>	<b>70°C</b>
0.3	4.81	3.24	0.3	n/a	n/a
0.5	13.70	7.10	0.5	38.00	17.54
0.6	23.13	10.50	0.6	78.65	31.11
1.0	187.72	50.36	1.0	n/a	n/a

## Other Data (PLCC)

Theta Ja =	61°C/W	Temp with Theta Ja Degree Kelvin	
Theta Jc =	20°C/W		
V <sub>CC</sub> =	5.25V	T(55°C) =	353.62 K
I <sub>CC</sub> @ 55°C =	80 mA	T(70°C) =	368.62 K
I <sub>CC</sub> @ 70°C =	80 mA	T(125°C) =	422.02 K
I <sub>CC</sub> @ 125°C =	75 mA	T(140°C) =	413 K
Boltzman's Constant = K = $8.62 \times 10^{-5}$ eV/K			

## Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLТ)			Bake (140°C)		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	4.92	3.30	0.3	n/a	n/a
0.5	14.25	7.31	0.5	38.00	17.54
0.6	24.25	10.89	0.6	78.65	31.11
1.0	203.16	53.51	1.0	n/a	n/a

Voltage Acceleration Factor (VAF) for HVLТ on this process is 55.

**NOTE:**

FIT = Failure in Time. 1 FIT = 1 failure per  $1 \times 10^9$  device hours.

**Failure Analysis**

- A. 1 oxide defect — 0.3 eV  
B. 2 oxide defects — 0.3 eV  
1 metal defect — 0.5 eV

- C. 2 oxide defects — 0.3 eV  
D. 1 passivation defect — 0.5 eV  
E. 1 passivation defect — 0.5 eV

## 5C090

Device: 5C090  
 Organization: 24 Macrocells  
 Pinout: 40 Lead, 600 mil Cerdip and PDIP (44 Lead PLCC)  
 Die Size: 166 x 181 mils  
 Transistor Count: ~ 30K  
 Process: CHMOS IIE, P424  
 Programming Voltage: 12.5V  
 Technology: CMOS

Table 1. Reliability Data

Package	Burn-In 48 Hours	125°C Dynamic Lifetest		
		168 Hours	500 Hours	1K Hours
CERDIP				
1987	1/2123	2/2085	0/170	0/170
1989	3/3934	3/3931	0/190	0/190
1991	0/2400	0/2399		
1992	0/1200	0/1196		
TOTALS	4/9657	5/9611	0/360	0/360
Additional Readouts: CERDIP 0/265 @ 2K Hours LT (1987 and 1989 material)				
Package	125°C High Voltage Dynamic Lifetest (6.5V)			
	48 Hours	168 Hours	500 Hours	
CERDIP				
1987	0/170	1/170	0/156	
1989	0/195	0/195	0/195	
1991	0/399	0/399	0/399	
1992	0/200	0/100	0/100	
TOTALS	0/964	1/864 (C)	0/850	
Package	Data Retention Bake (CERDIP @ 250°C)			
	48 Hours	168 Hours	500 Hours	
CERDIP				
1987	0/66	0/66	0/66	
1989	0/99	0/99	0/99	
TOTALS	0/165	0/165	0/165	

Table 2. Additional Qualification Stress Information

Package	Temperature Cycling (Condition "C"—150°C to -65°C)		
	200 Cycles	500 Cycles	1K Cycles
CERDIP	0/149	0/141	1/67 D
Package	Thermal Shock (Condition "C"—150°C to -65°C)		
	50 Cycles	200 Cycles	500 Cycles
CERDIP	0/50	0/50	1/50 E
Package	85°C/85% Relative Humidity		
	168 Hours	500 Hours	1K Hours
PLCC	0/172	0/172	0/172
Additional Readouts PLCC 0/172 @ 2K Hours			
Package	Steam (121°C, 2 ATM)		
	96 Hours	168 Hours	
PDIP	0/153	0/153	
PLCC	3/474 F	0/471	
TOTALS	3/627	0/624	
Additional Readouts: PLCC 0/300 @ 500 Hours of Steam			

Table 3. Failure Rate Predictions (5C090-CERDIP-Fab 1)

Actual Device Hrs	Ea (eV)	Equivalent Device Hrs		# Fail	Fail Rate in FITs (60% UCL)	
		55°C	70°C		55°C	70°C
1.43E + 06 2.77E + 05	0.3 LT 0.3 HVL * VAF	7.54E + 06 8.04E + 07	4.95E + 06 5.28E + 07	0 1		
Total 0.3 eV Failures =				1	23.0	35.0
1.43E + 06 2.77E + 05	0.5 LT 0.5 HVL	2.28E + 07 4.43E + 06	1.13E + 07 2.20E + 06	0 0		
Total 0.5 eV Failures =				0	33.6	67.6
1.43E + 06 2.77E + 05	0.6 LT 0.6 HVL	3.97E + 07 7.70E + 07	1.72E + 07 3.33E + 06	2 0		
Total 0.6 eV Failures =				2	65.5	151.6
1.43E + 06 2.77E + 05	1.0 LT 1.0 HVL	3.64E + 08 7.07E + 07	8.99E + 07 1.74E + 07	0 0		
Total 1.0 eV Failures =				0	0.0	0.0
Combined Failure Rate (FITs) =					122.0	254.2
48 Hour BI Infant Mortality = $4/7257 = 0.0554\% = 722 \text{ DPM @ } 60\% \text{ Confidence}$						



Table 4. Failure Rate Predictions (CERDIP 5C090-Fab 4)

Actual Device Hrs	Ea (eV)	Equivalent Device Hrs		# Fail	Fail Rate in FITs (60% UCL)	
		55°C	70°C		55°C	70°C
4.3E + 5 2.3E + 5	0.3 LT 0.3 HVL * VAF	2.3E + 6 3.1E + 7	1.5E + 6 2.7E + 7			
Total 0.3 eV Failures =				0	0	0
4.3E + 5 2.3E + 5	0.5 LT 0.5 HVL	6.9E + 6 3.6E + 6	3.4E + 6 1.8E + 6			
Total 0.5 eV Failures =				0	87	176
4.3E + 5 2.3E + 5	0.6 LT 0.6 HVL	1.2E + 7 6.3E + 6	5.2E + 6 2.7E + 6			
Total 0.6 eV Failures =				0	0	0
4.3E + 5 2.3E + 5	1.0 LT 1.0 HVL	1.1E + 8 5.8E + 7	2.7E + 7 1.4E + 7			
Total 1.0 eV Failures =				0	0	0
Combined Failure Rate (FITs) =					87	176
48 Hour BI Infant Mortality = 0/4199 = 0.00% = 219 DPM @ 60% Confidence						

## Other Data (CERDIP)

Theta Ja =	36°C/W	Temp with Theta Ja	
Theta Jc =	13°C/W	Degree Kelvin	
V <sub>CC</sub> =	5.25V	T(55°C) =	345.01 K
I <sub>CC</sub> @ 55°C =	90 mA	T(70°C) =	360.01 K
I <sub>CC</sub> @ 70°C =	90 mA	T(125°C) =	413.12 K
I <sub>CC</sub> @ 125°C =	80 mA	T(250°C) =	523 K

Boltzman's Constant = K =  $8.62 \times 10^{-5}$  eV/K

## Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVL)			Bake (250°C)		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	5.27	3.46	0.3	n/a	n/a
0.5	15.96	7.93	0.5	727.61	335.91
0.6	27.78	11.99	0.6	2718.18	1075.15
1.0	254.79	62.82	1.0	n/a	n/a

# Other Data (PDIP)

Theta Ja =	48°C/W	Temp with Theta Ja Degree Kelvin	
Theta Jc =	16°C/W		
V <sub>CC</sub> =	5.25V	T(55°C) =	350.68 K
I <sub>CC</sub> @ 55°C =	90 mA	T(70°C) =	365.68 K
I <sub>CC</sub> @ 70°C =	90 mA	T(125°C) =	418.16 K
I <sub>CC</sub> @ 125°C =	80 mA	T(140°C) =	413 K

Boltzman's Constant =  $K = 8.62 \times 10^{-5} \text{ eV/K}$

# Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLT)			Bake (140°C)		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	4.96	3.30	0.3	n/a	n/a
0.5	14.41	7.31	0.5	38.00	17.54
0.6	24.56	10.89	0.6	78.65	31.11
1.0	207.54	53.47	1.0	n/a	n/a

# Other Data (PLCC)

Theta Ja =	48°C/W	Temp with Theta Ja Degree Kelvin	
Theta Jc =	16°C/W		
V <sub>CC</sub> =	5.25V	T(55°C) =	350.68 K
I <sub>CC</sub> @ 55°C =	90 mA	T(70°C) =	365.68 K
I <sub>CC</sub> @ 70°C =	90 mA	T(125°C) =	418.16 K
I <sub>CC</sub> @ 125°C =	80 mA	T(140°C) =	413 K

Boltzman's Constant =  $K = 8.62 \times 10^{-5} \text{ eV/K}$

# Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLT)			Bake (140°C)		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	4.96	3.30	0.3	n/a	n/a
0.5	14.41	7.31	0.5	38.00	17.54
0.6	24.56	10.89	0.6	78.65	31.11
1.0	207.54	53.47	1.0	n/a	n/a

Voltage Acceleration Factor (VAF) for HVLT on this process is 55.

## NOTE:

FIT = Failure in Time. 1 FIT = 1 failure per  $1 \times 10^9$  device hours.

# Failure Analysis

- A. 1 SBCL — 0.6 eV
  - 1 polysilicon defect — 0.5 eV
  - 1 metal defect — 0.5 eV
  - 1 oxide defect — 0.3 eV
- B. 2 MBCL — 0.6 eV

- C. 1 oxide defect — 0.3 eV
- D. 1 open metal — 0.5 eV
- E. 1 hermeticity
- F. 3 passivation defects — 0.5 eV

# 5C180

Device: 5C180  
 Organization: 48 Macrocells  
 Pinout: 68 Lead PLCC, and PGA  
 Die Size: 265 x 182 mils  
 Transistor Count: ~ 60K  
 Process: CHMOS IIE, P424  
 Programming Voltage: 12.5V  
 Technology: CMOS

Table 1. Reliability Data

Package	Burn-In 48 Hours	125°C Dynamic Lifetest		
		168 Hours	500 Hours	1K Hours
CERQUAD	0/378	0/376	0/102	0/102
TOTALS	0/378	0/376	0/102	0/102

Package	125°C High Voltage Dynamic Lifetest (6.5V)		
	48 Hours	168 Hours	500 Hours
CERQUAD	0/50	0/43	0/39
TOTALS	0/50	0/43	0/39

Package	Data Retention Bake (CERDIP @ 250°C)		
	48 Hours	168 Hours	500 Hours
CERQUAD	0/107	0/100	0/100
TOTALS	0/107	0/100	0/100

Table 2. Additional Qualification Stress Information

Package	Temperature Cycling (Condition "C"—150°C to -65°C)		
	200 Cycles	500 Cycles	1K Cycles
PGA	0/128	0/128	0/128
PLCC	0/123	0/123	0/123
TOTALS	0/251	0/251	0/251

Package	Thermal Shock (Condition "C"—150°C to -65°C)		
	50 Cycles	200 Cycles	500 Cycles
PGA	0/117	0/117	0/177
TOTALS	0/117	0/117	0/177

Package	Steam (121°C, 2 ATM)	
	96 Hours	168 Hours
PLCC	0/77	0/77
TOTALS	0/77	0/77

Additional Readouts:	
PLCC	0/77 @ 500 Hours of Steam

Table 3. Failure Rate Predictions (5C180-CERQUAD)

Actual Device Hrs	Ea (eV)	Equivalent Device Hrs		# Fail	Fail Rate in FITs (60% UCL)	
		55°C	70°C		55°C	70°C
1.30E + 05 5.03E + 04	0.3 LT 0.3 HVLT * VAF	5.81E + 05 1.24E + 07	3.94E + 05 8.39E + 06	0 0		
Total 0.3 eV Failures =				0	70.6	104.2
1.30E + 05 5.03E + 04	0.5 LT 0.5 HVLT	1.58E + 06 6.11E + 05	8.25E + 05 3.20E + 05	0 0		
Total 0.5 eV Failures =				0	0.0	0.0
1.30E + 05 5.03E + 04	0.6 LT 0.6 HVLT	2.60E + 06 1.01E + 06	1.19E + 06 4.62E + 05	0 0		
Total 0.6 eV Failures =				0	0.0	0.0
1.30E + 05 5.03E + 04	1.0 LT 1.0 HVLT	1.91E + 07 7.41E + 06	5.24E + 06 2.03E + 06	0 0		
Total 1.0 eV Failures =				0	0.0	0.0
Combined Failure Rate (FITs) =					70.6	104.2
48 Hour BI Infant Mortality = $0/378 = 0.0000\% = 2425 \text{ DPM @ } 60\% \text{ Confidence}$						

3

## Other Data (CERQUAD)

Theta Ja =	42°C/W	Temp with Theta Ja	
Theta Jc =	14°C/W	Degree Kelvin	
V <sub>CC</sub> =	5.25V	T(55°C) =	358.87 K
I <sub>CC</sub> @ 55°C =	140 mA	T(70°C) =	373.87 K
I <sub>CC</sub> @ 70°C =	140 mA	T(125°C) =	424.46 K
I <sub>CC</sub> @ 125°C =	120 mA	T(250°C) =	523 K

Boltzman's Constant =  $K = 8.62 \times 10^{-5} \text{ eV/K}$ 

## Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLT)			Bake (250°C)		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	4.47	3.03	0.3	n/a	n/a
0.5	12.14	6.35	0.5	727.61	335.91
0.6	19.99	9.19	0.6	2718.18	1075.15
1.0	147.28	40.29	1.0	n/a	n/a



## Other Data (PLCC)

Theta Ja =	38°C/W	Temp with Theta Ja	
Theta Jc =	14°C/W	Degree Kelvin	
V <sub>CC</sub> =	5.25V	T(55°C) =	355.93 K
I <sub>CC</sub> @ 55°C =	140 mA	T(70°C) =	370.93 K
I <sub>CC</sub> @ 70°C =	140 mA	T(125°C) =	421.94 K
I <sub>CC</sub> @ 125°C =	120 mA	T(140°C) =	413 K

Boltzman's Constant =  $K = 8.62 \times 10^{-5} \text{ eV/K}$

## Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLT)			Bake (140°C)		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	4.61	3.11	0.3	n/a	n/a
0.5	12.78	6.62	0.5	38.00	17.54
0.6	21.28	9.65	0.6	78.65	31.11
1.0	163.38	43.76	1.0	n/a	n/a

## Other Data (PGA)

Theta Ja =	28.5°C/W	Temp with Theta Ja	
Theta Jc =	4°C/W	Degree Kelvin	
V <sub>CC</sub> =	5.25V	T(55°C) =	348.95 K
I <sub>CC</sub> @ 55°C =	140 mA	T(70°C) =	363.95 K
I <sub>CC</sub> @ 70°C =	140 mA	T(125°C) =	415.96 K
I <sub>CC</sub> @ 125°C =	120 mA	T(250°C) =	523 K

Boltzman's Constant =  $K = 8.62 \times 10^{-5} \text{ eV/K}$

## Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLT)			Bake (250°C)		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	4.98	3.30	0.3	n/a	n/a
0.5	14.53	7.33	0.5	727.61	335.91
0.6	24.82	10.91	0.6	2718.18	1075.15
1.0	211.14	53.68	1.0	n/a	n/a

Voltage Acceleration Factor (VAF) for HVLT on this process is 55.

## NOTE:

FIT = Failure in Time. 1 FIT = 1 failure per  $1 \times 10^9$  device hours.

## APPENDIX A

Failure Rate Calculations for  
60% Upper Confidence Level

- Step 1.** Collect burn-in and lifetest data for each lot after 48 hours of burn-in through lifetest for each lot.
- Step 2.** Determine the failure mechanism and assign an activation energy ( $E_A$ ) for each failure, except those occurring during the first 48 hrs.

**Failure Mechanism Activation Energies  
Relevant to EPROMs**

Failure Mode	Activation Energy
Defective bit charge gain/loss	0.6 eV
Oxide breakdown	0.3 eV
Silicon defects	0.3 eV
Contamination	1.0 eV–1.2 eV
Intrinsic charge loss	1.4 eV

**Step 3.** Calculate the total number of device hours accumulated beyond 48 hours of burn-in.

**NOTE:**

The first 48 hours of burn-in at either 5.25V or 6.5V measure infant mortality and are not included in the failure rate calculation. Monitor lots will use only 5.25V data for the infant mortality evaluation (IME). See monitor flow chart, Figure 1.

Example: 125°C Burn-In/Lifetest for a 2 lot sample

$$\frac{\text{\# failures}}{\text{total \# devices}}$$

	48 Hours	168 Hours	500 Hours	1K Hours	2K Hours
Lot #1	0/1000	1/1000	0/999	0/998	0/994
Lot #2	0/221	0/201	1/201	1/100	0/99
Totals	0/1221	1/1201	1/1200	1/1098	0/1093

Actual Device Hours =  $\Sigma$  (Number of Devices in Stress Interval) (Number of Hours in Stress Interval)

$$\begin{aligned}
 &= 1201 (168 \text{ hrs} - 48 \text{ hrs}) + 1200 (500 \text{ hrs} - 168 \text{ hrs}) \\
 &\quad + 1098 (1000 \text{ hrs} - 500 \text{ hrs}) + 1093 (2000 \text{ hrs} - 1000 \text{ hrs}) \\
 &= 1201 (120 \text{ hrs}) + 1200 (332 \text{ hrs}) + 1098 (500 \text{ hrs}) \\
 &\quad + 1093 (1000 \text{ hrs}) \\
 &= 2.185 \times 10^6 \text{ Device Hours}
 \end{aligned}$$

mechanism), or use the Arrhenius relation.

$$R = A \exp \left[ \frac{-E_A}{KT} \right]$$

$K = 8.617 \times 10^{-5} \text{ eV/}^\circ\text{K}$  (Boltzmann's constant)

$A$  = proportionality constant

$R$  = mean rate to failure

$E_A$  = activation energy

$T$  = temperature in Kelvin

$$\frac{R_1}{R_2} = \frac{A_1 \exp \left[ \frac{-E_A}{KT_1} \right]}{A_2 \exp \left[ \frac{-E_A}{KT_2} \right]} = \exp \left[ \left( \frac{E_A}{K} \right) \left( \frac{1}{T_2} - \frac{1}{T_1} \right) \right]$$

Where  $A_1 = A_2 = A$  for the same failure mechanism (i.e., same  $E_A$ )

Where  $R_1$  and  $R_2$  are rates for a normal operating temp and an elevated temperature respectively.

$$R_1 = R_2 \times \exp \left[ \left( \frac{E_A}{K} \right) \left( \frac{1}{T_2} - \frac{1}{T_1} \right) \right]$$

However, since rate ( $R$ ) has the units  $1/\text{time}$ , we can think in terms of time to one failure or MTBF.

Thus:

$$R_1 = \frac{1}{t_1} \text{ where } t_1 = \text{MTBF at some temperature } T_1$$

and:

$$R_2 = \frac{1}{t_2} \text{ where } t_2 = \text{MTBF at some temperature } T_2$$

Thus the Arrhenius relation becomes:

$$\frac{1}{t_1} = \frac{1}{t_2} \times \exp \left[ \frac{E_A}{K} \left( \frac{1}{T_2} - \frac{1}{T_1} \right) \right]$$

or:

$$t_1 = \exp \left[ \frac{E_A}{K} \left( \frac{1}{T_1} - \frac{1}{T_2} \right) \right] \times t_2$$

We then define the Acceleration Factor as:

$$\text{A.F.} = \frac{t_1}{t_2} = \exp \left[ \frac{E_A}{K} \left( \frac{1}{T_1} - \frac{1}{T_2} \right) \right]$$

For example: For  $E_A = 0.6 \text{ eV}$ ,  $T_2 = 398^\circ\text{K}$ ,  $T_1 = 328^\circ\text{K}$

$$t_1 = 41.7 t_2$$

Therefore, one hour at  $125^\circ\text{C}$  is equivalent to 41.7 hours at  $55^\circ\text{C}$  for a failure mechanism of activation energy  $E_A = 0.6 \text{ eV}$ . Then 41.7 is the thermal acceleration factor for time.

**NOTE:**

The Arrhenius Plot (Figure 2, Page 3) is simply  $\ln$  (Acceleration Factor) vs.  $1/\text{Temperature}$  normalized for an MTBF of one hour at  $250^\circ\text{C}$  ( $T_2$ ). This plot can also be used to determine the acceleration factor between two temperatures other than  $250^\circ\text{C}$ .

For example: For a 0.3 eV failure at  $125^\circ\text{C}$ , the acceleration factor is 8.1 relative to a 0.3 eV failure at  $250^\circ\text{C}$ . For a 0.3 eV failure at  $25^\circ\text{C}$ , the acceleration factor is 152 relative to  $250^\circ\text{C}$ . Therefore, the acceleration factor between  $125^\circ\text{C}$  and  $25^\circ\text{C}$  is:

$$\text{A.F.} = \frac{t_1}{t_2} = \frac{152}{8.1} = 18.7$$

**Step 5.** Organize the burn-in/lifetest data by  $E_A$ , Total Device Hours at the burn-in/lifetest temperature  $T_2$ , Thermal Acceleration Factors for each failure mechanism ( $E_A$ ), Number of Failures for each failure mechanism, and the calculated equivalent device hours at the desired operating temperature  $T_1$ .

**NOTE:**

The rise in junction temperature due to the thermal resistivity of the package ( $\theta_{JA}$ ) must be added to the ambient temperature to arrive at the actual burn-in/lifetest temperatures.

$$T_{\text{test}} = T_J + T_{\text{Ambient}} = \theta_{JA} (IV @ T_{\text{Ambient}}) + T_{\text{Ambient}}$$

$E_A$ (eV)	Total Device Hrs @ $T_2$	Acceleration Factors	#Fail	Equivalent Hours @ $T_1$
0.3	T.D.H.	X	$N_1$	X (T.D.H.)
0.6	T.D.H.	Y	$N_2$	Y (T.D.H.)
1.0	T.D.H.	Z	$N_3$	Z (T.D.H.)

The failure rates for individual failure mechanisms and the total combined failure rate can be predicted using the data table and the following formula:

$$\text{FITs} = \frac{\chi^2(n, \alpha)}{2T} \left( 10^9 \right)$$

Where  $\chi^2(n, \alpha)$  is the value of the chi-squared distribution for  $n$  degrees of freedom and confidence level of  $\alpha$ . The degrees of freedom,  $n = [2(\# \text{ of failures}) + 2]$  for this application.  $T$  is the total equivalent device hours at  $T_1$ . The total combined failure rate is just the sum of the individual failure rates for each failure mechanism.

For a 60% UCL (Upper Confidence Limit), the above formula converts to the following:

#Failures	FIT Rate (60% UCL)
0	$0.915 \times 10^9/T$
1	$2.02 \times 10^9/T$
2	$3.105 \times 10^9/T$
3	$4.17 \times 10^9/T$
$3 < \# < 15$	$\left[ \frac{1.049 (\# \text{failures for a particular } E_A) + 1.0305}{T} \right] \left[ 10^9 \right]$
$> 15$	$\left[ \frac{0.2533 + \sqrt{(4 \times \# \text{Failed}) + 3}}{4T} \right] \left[ 10^9 \right]$



## Example 1:

Assume for this example, that  $I_{CC}$  active is 57 mA at  $T_{Ambient} = 125^{\circ}\text{C}$  and  $I_{CC}$  active is 60 mA at  $T_{Ambient} = 55^{\circ}\text{C}$ .

Also assume that  $\theta_{JA} = 35^{\circ}\text{C/W}$ .

Then,

$$T_2 = (35^{\circ}\text{C/W}) (57 \text{ mA}) (5\text{V}) + 125^{\circ}\text{C} \\ \cong 135^{\circ}\text{C} = 408^{\circ}\text{K}$$

$$T_1 = (35^{\circ}\text{C/W}) (60 \text{ mA}) (5\text{V}) + 55^{\circ}\text{C} \\ \cong 65^{\circ}\text{C} = 338^{\circ}\text{K}$$

$E_A$ (eV)	Actual Device Hours @ $125^{\circ}\text{C}$	Acceleration Factors For $135^{\circ}\text{C}$ to $65^{\circ}\text{C}$	Equivalent Hours at $55^{\circ}\text{C}$	# Fail	$55^{\circ}\text{C}$ FIT Rate
0.3	$2.185 \times 10^6$	5.85	$1.278 \times 10^7$	0	81
0.6	$2.185 \times 10^6$	34.18	$7.468 \times 10^7$	2	42
1.0	$2.185 \times 10^6$	359.93	$7.864 \times 10^8$	1	3
Total Combined Failure Rate =					126 FITs

## Example 2:

Assume than an additional lot of 800 CHMOS III-E devices is burned in using a 6.5V lifetest as shown below. Assume further that the one failure shown at 168 hours is a 0.3 eV oxide failure. Using Table 2 below, a voltage acceleration factor of 26 results from a 1.25V voltage overstress (5.25V to 6.5V).

	48 Hours	168 Hours	500 Hours
Lot #3	0/800	1/800	0/799

$$\text{Actual Device Hours} = 800 (48 \text{ hrs} - 0 \text{ hrs}) + 800 (168 \text{ hrs} - 48 \text{ hrs}) + 799 (500 \text{ hrs} - 168 \text{ hrs}) \\ = 3.997 \times 10^5$$

Table 2. Time-Dependent Oxide Failure Voltage Accelerations Relative to 5.25V

Type	Supply Voltage (Volts)	Oxide Thickness (Å)	Operating Stress (MV/cm)	Lifetest Stress Voltage			
				5.5V	6.0V	6.5V	7.0V
CHMOS III-E	5	235	2.15	1.9	7.0	26	93

## ASSUMES:

- Failure rate calculations use the appropriate acceleration factor for stress voltage versus 5.25V operating voltage (conservative).
- Reference [2] E. Nelson Anolick.

Since this voltage accelerated stress is used to predict an oxide breakdown failure rate, the 5.25V burn-in/lifetest 55°C equivalent hours for  $E_A = 0.3$  eV are added to the 6.5V burn-in/lifetest 55°C equivalent hours as follows:

125°C Burn-In/Lifetest	$E_A$ (eV)	Actual Device Hours @ 125°C	Acceleration Factors for 135°C to 65°C	Equivalent Hours @55°C
5.25V	0.3	$2.185 \times 10^6$	5.85	$1.278 \times 10^7$
6.5V	0.3	$3.997 \times 10^5$	(5.85 x 26)	$6.079 \times 10^7$
Total Equivalent Device Hours for 0.3 eV Failures = $7.357 \times 10^7$				

The following failure rate predictions include the total equivalent 55°C,  $E_A = 0.3$  eV device hours found above:

$E_A$ (eV)	Actual Device Hours @ 125°C	Acceleration Factors for 135°C to 65°C	Equivalent Hours @55°C	# Fail	55°C FIT Rate
0.3 ELT 0.3 HVELT	$2.185 \times 10^6$ $3.997 \times 10^5$	5.85 (5.85 x 26)	$7.357 \times 10^7$	1	27
0.6 ELT 0.6 HVELT	$2.185 \times 10^6$ $3.997 \times 10^5$	34.18 34.18	$8.834 \times 10^7$	2	35
1.0 ELT 1.0 HVELT	$2.185 \times 10^6$ $3.997 \times 10^5$	359.93 359.93	$9.303 \times 10^8$	1	2
Total Combined Failure Rate =					84 FITs

**NOTES:**

1. Additional information on calculating failure rates is contained in the April 2, 1984 International Reliability Physics Symposium editorial entitled "Calculating Failure Rates from Stress Data" by Robert M. Alexander.
2. 1 FIT = 1 Failure Unit = 0.0001%/1K hours.



---

# Development Support Tools

---

4







# PLD SUPPORT AT A GLANCE

Device		Intel		Data I/O*				Logical Devices	
PLD	Packages	PLDshell Plus S/W	GUPI Adaptor	ABEL* S/W	UNISITE*	Model 2900	Model 29B LogicPak*	CUPL S/W	ALLPRO 40/88
85C220-80/66	D, P N, M	V2.1	20D20J	V3.2	SITE40/48 (V2.8) CHIPSITE (V3.0)	V1.1 V1.1	303A-011A (V10) 303A-011B (V05)	V3.2	1.50c 1.50c
85C220-7/10	N	V2.1	20D20J	V3.2	CHIPSITE (V3.0)	V1.1	303A-011B (V05)	V3.2	1.50c
85C224-80/66	D, P N, M	V2.1	24D28J	V4.0	SITE40/48 (V3.1) CHIPSITE (V3.1)	V1.2 V1.2	303A-011A (V14) 303A-011B (V06)	V4.0	2.1 2.1
85C224-7/10	N	V2.1	24D28J	V4.0	CHIPSITE (V3.1)	V1.2	303A-011B (V06)	V4.0	2.1
85C22V10	D, P N	V2.1	22V10	V4.2	SITE40/48 (V3.7) CHIPSITE (V3.7)	V1.8 V1.8	303A-011A (V18) 303A-011B (V08)	V4.0	
85C060	D, P N, M	V2.1	LOGIC IID(1)	V3.0	SITE40/48 (V3.2) CHIPSITE (V3.2)	V1.5 V1.5	303A-011A (V14) 303A-011B (V06)	V2.15	2.1 2.1
85C090	D, P N	V2.1	LOGIC IID(2)	V3.0	SITE40/48 (V3.2) CHIPSITE (V3.2)	V1.5 V1.5	303A-010 (V03) 303A-010 (V03)	V2.50	2.1 2.1
85C508	D, P N	V2.1	85EPLD28	V3.2	SITE40/48 (V3.3) CHIPSITE (V3.4)	V1.5 V1.5	303A-011B (V06) 303A-011B (V06)	V3.2 V4.2(3)	2.1(3) 2.1(3)
5AC312	D, P N, M	V2.1	LOGIC IID(1)	V4.0	SITE40/48 (V2.7) CHIPSITE (V3.0)	V1.0 V1.1	303A-011A (V07) 303A-011B (V06)	V3.2	V1.48 V1.48
5AC324	D, P N	V2.1	40D44J	V3.2	SITE40/48 (V3.4) PINSITE (V3.4)	V1.5 V1.5	n/s n/s	V3.2	V1.50 V1.50
5C032	D, P	V2.1	20D20J	V3.0	SITE40/48 (V2.7)	V1.0	303A-011A (V02)	V2.15	V1.46



Model 3900: All devices supported with V1.2 (now)

## NOTES:

1. PLCC requires an ADAPT24TO28 in addition to the LOGIC IID.
2. PLCC requires an ADAPT40TO44 in addition to the LOGIC IID.
3. Projected support; call vendor for current information.
4. Complete support provided as 85C22V10.
5. Complete support provided as 85C060.
6. Complete support provided as 85C090.

n/s = Will not be supported due to hardware limitation.

\*Data I/O = 1-800-3-DATAIO

Logical Devices = 1-800-331-7766

Device		Intel		Data I/O*				Logical Device	
PLD	Packages	PLDshell Plus S/W	GUPI Adaptor	ABEL* S/W	UNISITE*	Model 2900	Model 29B LogicPak*	CUPL S/W	ALL 40/
5C060	D, P N, M	V2.1	LOGIC IID(1)	V3.0	SITE40/48 (V2.7) CHIPSITE (V2.7)	V1.0 V1.1	303A-011A (V01) 303A-011B (V01)	V2.15	V1. V1.
5C090	D, P N, M	V2.1	LOGIC IID(2)	V3.0	SITE40/48 (V2.7) CHIPSITE (V2.7)	V1.0 V1.1	303A-010 (V03) 303A-010 (V03)	V2.50	V1. V1.
5C180	N, M	V2.1	LOGIC 18	V3.0	CHIPSITE (V2.7)	n/s	n/s	V3.0	V1.
PLD16V8XP-7/10	P N	V2.1	20D20J	V3.2	SITE40/48 (V3.8) CHIPSITE (V3.8)	V1.9 V1.9	n/s n/s	Compile as 16V8	Corr as 1
PLD20V8XP-7/10	P N	V2.1	24D28J	V4.0	SITE40/48 (V3.8) CHIPSITE (V3.8)	V1.9 V1.9	n/s n/s	Compile as 20V8	Corr as 2
PLD22V10	P N	V2.1	22V10	V4.0(4)	SITE40/48 (V3.8) CHIPSITE (V3.8)	V1.9 V1.9	303A-010 (V18) 303A-010 (V08)(4)	V4.0(4)	2; 2;
PLD610	P N	V2.1	LOGIC IID(1, 5)	V3.0(5)	SITE40/48 (V3.8) CHIPSITE (V3.8)	V1.9 V1.9	303A-011A (V14) 303A-011B (V06)(5)	V2.15(5)	2; 2;
PLD910	P N	V2.1	LOGIC IID(2, 6)	V3.0(6)	SITE40/48 (V3.8) CHIPSITE (V3.8)	V1.9 V1.9	303A-010 (V03) 303A-010 (V03)(6)	V2.50(6)	2; 2;

**NOTES:**

1. PLCC requires an ADAPT24TO28 in addition to the LOGIC IID.
2. PLCC requires an ADAPT40TO44 in addition to the LOGIC IID.
3. Projected support; call vendor for current information.
4. Complete support provided as 85C22V10
5. Complete support provided as 85C060
6. Complete support provided as 85C090

n/s = Will not be supported due to hardware limitation.

\*Data I/O = 1-800-3-DATAIO

Logical Devices = 1-800-331-7766

ABEL, UNISITE, and LogicPak are trademarks of Data I/O, Corporation.

CUPL and ALLPRO are trademarks of Logical Devices, Inc.

↓  
Model 3900: All devices supported with V1.2 (now)

## Development Tools Overview

PAGE	CONTENTS	PAGE	CONTENTS
4-8	ZEELAN TECHNOLOGY, INC.	4-8	PROGRAMMABLE LOGIC SOFTWARE
4-8	ModelStation	4-8	VENDORS
4-8	PROGRAMMABLE LOGIC	4-8	ALCORN
4-8	PROGRAMMING VENDORS	4-8	ATGEM Test Generation
4-8	ADVIN	4-8	CADENCE
4-8	Solar PAL5A and Solar PAL5B	4-8	COMPUSER
4-8	BR MICROSYSTEMS	4-8	1 VHDL-XLTM
4-8	PLD-1100	4-8	DATA INCORPORATION
4-8	BYTEK CORPORATION	4-8	ASEL P-4
4-8	INSTRUMENT SYSTEMS DIVISION	4-8	PLDtest Plus
4-8	128H-U and 128H-L	4-8	INTEL CORPORATION
4-8	DATA I/O CORPORATION	4-8	PLDtest Plus
4-8	Unitatm	4-8	LOGIC MODELING
4-8	Model 2900	4-8	CORPORATION
4-8	DIGELLEC INC.	4-8	Strength Library
4-8	ELAN	4-8	LOGIC
4-8	Model 6000	4-8	CUDA
4-8	INTEL CORPORATION	4-8	MENTOR GRAPHICS
4-8	IUP-PC and IUP-200A-IUP-201A	4-8	Item Station
4-8	LOGICAL DEVICES, INC.	4-8	Design Architect
4-8	ALLPORT	4-8	Autologic
4-8	SMS	4-8	ClickSim
4-8	Sprint Plus	4-7	MHC INCORPORATED
4-8	STAG MICROSYSTEMS, INC.	4-7	PLD Designer and PLD Designer-XL
4-8	ZL-30, ZL-30A, ZL-33, PZ	4-7	O-CAD
4-8	THIRD-PARTY SUPPORT SUMMARY	4-7	PLD Tools (O-CAD/PLD)
4-7	PLD Programming Support	4-7	Stratistic Design Tool
4-7	PLD Programming Vendors	4-7	O-CAD (BET)
4-7	MARKETING ENGINEER	4-7	Versatile Simulation Tool
4-7	MARKETING ENGINEER	4-7	CHICK-VST
4-7	MARKETING ENGINEER	4-7	QUAD DESIGN
4-7	MARKETING ENGINEER	4-7	MOVIEW
4-7	MARKETING ENGINEER	4-7	VIEWLOGIC
4-7	MARKETING ENGINEER	4-7	View PLD and PowerView

November 1992

## Development Tools Overview

**MARYANN L. LINDQUIST**  
MARKETING ENGINEER



## Development Tools Overview

CONTENTS	PAGE	CONTENTS	PAGE
<b>PROGRAMMABLE LOGIC SOFTWARE VENDORS</b> .....	4-5	<b>ZEELAN TECHNOLOGY, INC.</b> .....	4-8
<b>ACUGEN</b> .....	4-5	ModelStation .....	4-8
ATGEN™ Test Generation .....	4-5	<b>PROGRAMMABLE LOGIC PROGRAMMING VENDORS</b> .....	4-9
<b>CADENCE</b> .....	4-5	<b>ADVIN</b> .....	4-9
Composer™ .....	4-5	Sailor PAL/SA and Sailor PAL/SB .....	4-9
Verilog-XL™ and VHDL-XL™ .....	4-5	<b>BP MICROSYSTEMS</b> .....	4-9
<b>DATA I/O CORPORATION</b> .....	4-5	PLD-1100 .....	4-9
ABEL®-4 .....	4-5	<b>BYTEK CORPORATION, INSTRUMENT SYSTEMS DIVISION</b> ...	4-9
PLDtest™ Plus .....	4-5	135H-U and 145H-U .....	4-9
<b>INTEL CORPORATION</b> .....	4-5	<b>DATA I/O CORPORATION</b> .....	4-9
PLDshell Plus .....	4-5	Unisite™ .....	4-9
<b>LOGIC MODELING CORPORATION</b> .....	4-6	Model 2900 .....	4-9
SmartModel Library .....	4-6	<b>DIGELEC INC.</b> .....	4-9
<b>LOGICAL DEVICES, INC.</b> .....	4-6	Uniport .....	4-9
CUPL™ .....	4-6	<b>ELAN</b> .....	4-9
<b>MENTOR GRAPHICS</b> .....	4-6	Model 6000 .....	4-9
Idea Station .....	4-6	<b>INTEL CORPORATION</b> .....	4-10
Design Architect .....	4-6	iUP-PC and iUP-200A/iUP-201A .....	4-10
Autologic .....	4-6	<b>LOGICAL DEVICES, INC.</b> .....	4-10
QuickSim II .....	4-6	ALLPRO™ .....	4-10
<b>MINC INCORPORATED</b> .....	4-7	<b>SMS</b> .....	4-10
PLDesigner and PLDesigner-XL .....	4-7	Sprint Plus .....	4-10
<b>OrCAD</b> .....	4-7	<b>STAG MICROSYSTEMS, INC.</b> .....	4-10
PLD Tools (OrCAD/PLD) .....	4-7	ZL-30, ZL-30A, ZL33, PPZ .....	4-10
Schematic Design Tool (OrCAD/DST) .....	4-7	<b>THIRD-PARTY SUPPORT SUMMARY</b> ..	4-11
Verification, Simulation Tool (OrCAD/VST) .....	4-7	PLD Programming Support .....	4-11
<b>QUAD DESIGN</b> .....	4-7	PLD Programming Vendors .....	4-13
MOTIVETM .....	4-7	PLD Software Support .....	4-15
<b>VIEWLOGIC</b> .....	4-7	PLD Software Vendors .....	4-16
ViewPLD and PowerView .....	4-7		

## DEVELOPMENT TOOLS OVERVIEW

### Programmable Logic Software Vendors

#### ACUGEN

##### ATGEN™ Test Generation

Automatically generates high-coverage functional test vectors for all types of Programmable Logic Devices. These test vectors can be applied after programming either on the programmer, a device tester, or during in-circuit testing.

- Event driven time-based simulator, concurrent fault simulation.
- Devices supported: PLDs and FPGAs.
- Testers supported: Sentry, GenRad, Teradyne, HP, and others.
- MS-DOS, DEC VAX/VMS, Sun-3, and Sun-4.

427-3 Amherst Street, Suite 391, Nashua, NH 03063 603-891-1995

#### CADENCE

##### Composer™

A comprehensive suite of design entry, debug, and documentation capabilities.

- Top-down, mixed-level design allows creation of hierarchical schematics mixed with textual design descriptions.
- Advanced editing capabilities minimize tedious entry tasks, freeing time for design analysis.
- Sophisticated, on-line rule checks for detecting design errors.
- Integration provides a tightly-coupled solution, binding together leading applications for analysis, synthesis, layout, and documentation.

PLD support is also available through integration of Data I/O and Minc PLD tools.

##### Verilog-XL™ and VHDL-XL™

Digital logic simulators provide advanced simulation capabilities to handle complex electronic designs.

- High-performance simulation meets difficult design challenges.

- Mixed-level simulation.
- Complete interactive debug environment.
- Accurate modeling and extensive library support.
- Open design environment.

555 River Oaks Parkway, Bldg. 1, San Jose, CA 95134 408-943-1234

#### DATA I/O CORPORATION

##### ABEL®-4

Design software lets you describe and implement logic designs using behavioral language and can be simulated and converted into JEDEC standard format.

- Devices supported: PLD, FPGA, and PROM.
- Entry methods: ABEL, Truth Table, State Machine, and Boolean Equation.
- MS-DOS, Sun-3, SunSPARC, DEC VAX/VMS, Apollo™.

##### PLDtest™ Plus

An integrated software package that combines a testability analysis of the device under design or test with fault grading and automatic test vector generation (ATVG).

- MS-DOS, DEC VAX/VMS v5.0, Sun-3, and SunSPARC stations.

10525 Willows Road N.E., P.O. Box 97046, Redmond, WA 98073-9746 206-881-6444

#### INTEL CORPORATION

##### PLDshell Plus

Design software that provides an easy-to-use menu system for PLD and FPGA design. Features include:

- Design merge.
- Functional simulation.

- Disassemble/convert utilities for existing JEDEC files.

Please see the PLDshell Plus datasheet in this section for more information.

See Sales Office and Distributor Listings at Back of Handbook

## LOGIC MODELING CORPORATION

### SmartModel Library

Combines the benefits of behavioral models, including fast simulation and minimum storage requirements, with structural level accuracy.

- Behavioral models for high-performance simulation.
- Accurate modeling of device functionality and timing.
- Intelligent handling of unknown conditions.

Simulators supported:

- AT&T, DAZIX/Intergraph, Mentor Graphics, Valid Logic Systems, ViewLogic Systems, Cadence Design Systems, Genrad, Racal-Redac, and Vantage Analysis Systems

Platforms supported:

- Data General, HP/Apollo, Intergraph, Sun Microsystems, Digital Equipment Corporation, IBM, and Sony

19500 N.W. Gibbs Drive, P.O. Box 310, Beaverton, OR 97075 503-690-6900

## LOGICAL DEVICES, INC.

### CUPL™

A high-level, universal design software package for PLD's and FPGA's offering:

- Variety of design expression formats
- True language flexibility
- DeMorgan expansion
- Simulation

- Output JEDEC Format

1201 N.W. 65th Place, Ft. Lauderdale, FL 33309  
305-974-0967

## MENTOR GRAPHICS

### Idea Station

Includes the company's Design Architect design creation tools, a System 1076 VHDL editor, and the QuickSim II high-performance logic simulator. An additional option of the Idea Station is the Autologic family of logic synthesis tools, which help put a complete, top-down design methodology into every designer's hands.

### Design Architect

An integrated system of schematic, symbol, and text editors for capturing designs at both the architectural and detailed logic levels, using either schematics or VHDL descriptions. Features include:

- Schematic capture
- VHDL description
- Automatic net routing
- Interwindow editing
- HP/Apollo-Mentor workstations

### Autologic

Autologic family of logic synthesis tools, which help put a complete, top-down design methodology into every designer's hands.

### QuickSim II

High performance logic simulator for function and performance verification. QuickSim II supports a complete hierarchy of modeling methods, including gate and table primitives, behavioral, VHDL and hardware based models.

8005 S.W. Boeckman Road, Wilsonville, OR 97070-7777 503-685-7000

**MINC INCORPORATED****PLDesigner and PLDesigner-XL**

A powerful design tool can be used for all types of programmable logic including FPGA's and PLD's.

- Automatic device selection based on user-specified design criteria.
- Automatic partitioning across multiple PLDs.
- Quick, easy retargeting between FPGA and PLD.
- Functional simulation.
- Import from high-level language, waveform editor, or schematic import.

6755 Earl Drive, Colorado Springs, CO 80918  
719-590-1155

**OrCAD****PLD Tools (OrCAD/PLD)**

Software tool for programming PLDs and FPGAs. JEDEC Format output.

- Schematic entry from OrCAD SDT.
- Test vector generation.
- 7 forms of input: Boolean equations, State Machines, Truth Tables.
- MS-DOS, Sun

**Schematic Design Tool (OrCAD/DST)**

Graphical environment for electrical design projects.

- Import/export to PSpice, OrCAD/PLD, OrCAD/VST.
- Netlist conversion into 30 different formats including Intel ADF.
- MS-DOS, Sun

**Verification, Simulation Tool (OrCAD/VST)**

A series of software tools for performing timing-based simulation of digital designs.

- Includes component modeling program.

- Accepts OrCAD/SDT generated netlists
- MS-DOS, Sun

3175 N.W. Alcock Drive, Hillsboro, OR 97124  
508-690-9881

**QUAD DESIGN****MOTIVETM**

An advanced timing verifier. Identifies all setup and hold violations in a design, without test vectors, by exhaustively tracing every signal delay path.

- Tests designs ranging from ASICs and FPGAs to systems.
- Support for 85C22V10 and iPLD22V10.
- 386 and 486 PCs; Sun-3 and Sun-4; Apollo; DEC; HP; and IBM RS6000 workstations.

Support for 85C22V10 and iPLD22V10 available now.

1385 Del Norte Road, Camaillo, CA 93010  
805-988-8250

4

**VIEWLOGIC****ViewPLD and PowerView**

Integrated schematic capture and simulation environment.

- Entry methods: VHDL, Schematic, ABEL, JEDEC, State Machine, Truth Table, Logic Equations.
- Automatically partitioning and optimization into PLD architecture.
- Full timing/fault simulation.
- IBM PC—WorkView
- UNIX—PowerView

2350 Mission College Blvd, Santa Clara, CA 95054  
408-982-3881



## ZEELAN TECHNOLOGY, INC.

### ModelStation

Workstation system that makes it possible for designers to generate extremely accurate analog and mixed analog-digital models. Designers can quickly specify real-life operating conditions, characterize the device within that environment and automatically extract an accurate mathematical model based upon a comprehensive set of measurements of the real behavior of the device.

Simulation Compatibilities include:

- SPICE, and derivatives such as H-Spice, P-Spice, Analogy-Saber, Cadence-Spectre + Profile, Mentor Graphics-Accusim, QuadDesign and Quantic Laboratories signal integrity design/analysis products.

8305-D S.W. Creekside Place, Beaverton, OR  
97005 503-520-1000

## Programmable Logic Programming Vendors

### ADVIN

#### Sailor PAL/SA and Sailor PAL/SB

A universal programmer providing support for Intel DIP devices with expansion modules.

1050 E. Duane Ave, Suite L, Sunnyvale, CA 94086  
408-984-8600

### BP MICROSYSTEMS

#### PLD-1100

A universal programmer providing support for Intel DIP devices with socket converters from outside sources.

10681 Haddington, Suite 190, Houston, TX 77043  
800-225-2102 or 713-461-9430

### BYTEK CORPORATION, INSTRUMENT SYSTEMS DIVISION

#### 135H-U and 145H-U

A universal programmer providing support for Intel DIP devices.

543 N.W. 77th St, Boca Raton, FL 33487  
800-523-1565 or 407-994-3520

### DATA I/O CORPORATION

#### Unisite™

Device programmer

- RS232C port connection to PC.

- Devices supported: DIP, PGA, PLCC, SOIC, and LCC.
- Can program almost all PLDs and FPGAs, memory devices, and microcontrollers.
- File types: JEDEC, Intel Hex-32, and many others.

#### Model 2900

Device programmer

- RS-232-C port.
- Devices supported: DIP, PLCC, and LCC.
- Can program almost all PLDs and FPGAs, and memory devices.
- Testing: Functional, Parallel test vector, and device continuity.

10525 Willows Road N.E., P.O. Box 97046,  
Redmond, WA 98073-9746 206-881-6444

### DIGELEC INC.

#### Uniport

A universal programmer providing support for Intel DIP devices with socket adapters.

20144 Plummer St, Chatsworth, CA 91311  
800-367-8750 or 818-701-9677

### ELAN

#### Model 6000

A universal programmer. The PD40 can program DIP packages having up to 40 pins. The PD84 has a 48-pin DIP and a universal PLCC socket which can support up to 84 PLCC devices.

538 Valley Way, Milpitas, CA 95035  
800-541-3526 or 408-946-8495

## INTEL CORPORATION

## iUP-PC and iUP-200A/iUP-201A

Intel specific PC-based programmer with GUPI Adaptors. Intel specific standalone programmer with GUPI Adaptors.

See Sales Office and Distributor Listings at Back of Handbook

## LOGICAL DEVICES, INC.

## ALLPRO™

Universal device programmer.

- Bus card interface for PC, RS-232 connection for all other computers.
- Devices supported: DIP, PGA, PLCC, SOIC, and LCC device packages.
- Testing: Functional, Diagnostic, Internal Logic, and others.
- File types: JEDEC; Intel Hex 80, 86; and others.

1201 N.W. 65th Place, Ft. Lauderdale, FL 33309  
305-974-0967

## SMS

## Sprint Plus

A universal 28-pin programmer with optional expansion pods.

SMS GmbH, Im Grund 15, D-7988, WANGEN  
49-7522-5018

## STAG MICROSYSTEMS, INC.

## ZL-30, ZL-30A, ZL33, PPZ

Logic programmer and adaptors, gang programmer and universal programmer.

1600 Wyatt Drive, Suite 3, Santa Clara, CA 95054  
800-227-8836 or 408-988-1118

## Third-Party Support Summary

(Support claimed by manufacturer)

### PLD PROGRAMMING SUPPORT

Vendor	Product	Module	Adaptor/Device/Version/Etc.
Advin	Sailor PAL/SA Sailor PAL/SB	—	DIPs: 5C031, 5C032, 5C060
		—	(same as above)
		w/EM-900	DIP: 5C090
		w/EM-1800	PLCC: 5C180
		w/PLCC Adap.	PLCCs: 5C060, 5C090
BP Micro	PLD-1100	—	V1.46—DIPs: 85C220, 85C224, 5AC312, 5C031, 5C032, 5C060 w/socket converter from outside source—PLCCs: 85C220, 85C224, 5AC312, 5C060
	PLD-1128	—	V1.46—DIPs: 85C220, 85C224, 5AC312, 5C031, 5C032, 5C060 w/socket converter from outside source—PLCCs: 85C220, 85C224, 5AC312, 5C060
Bytek	135H-U	UNICEL	w/LTA00C—DIPs: 5C031, 5C032, 5C060, 5C090, 5C180, 5AC312, 85C220, 85C224
	145H-AD	—	DIPs: 5C031, 5C032, 5C060, 5C090, 5C180, 5AC312, 85C220, 85C224
Data I/O	UNISITE	—	V3.1—DIPs: 85C220, 85C224, 85C060, 85C508, 85C22V10, PLD610, PLD910, PLD22V10, 5AC312, 5AC324, 5C031, 5C032, 5C060, 5C090 V3.1—PLCCs: 85C220, 85C224, 85C060, 85C508, 85C22V10, PLD610, PLD910, PLD22V10, 5AC312, 5AC324, 5C060, 5C090, 5C180
	Model 2900	—	V1.3—DIPs: 85C220, 85C224, 85C060, 85C508, 5AC312, 5AC324, 5C031, 5C032, 5C060, 5C090 V1.3—PLCCs: 85C220, 85C224, 85C060, 85C508, 5AC312, 5AC324, 5C060, 5C090
	Model 29B	LogicPak V.4	303A-011A (V10)—DIPs: 85C220, 85C224, 85C060, 5AC312, 5C031, 5C032, 5C060 303A-011B (V05)—PLCCs: 85C220, 85C224, 85C060, 5AC312, 5C031, 5C032, 5C060 303A-010 (V03)—DIP & PLCC: 85C090, 5C090
	Model 60A/H	—	DIPs: 5C031, 5C032, 5C060
Digelec	860	—	DIPs: 5C031, 5C032, 5C060, 5C090
		w/8601	PLCCs: 5C060, 5C090, 5C180
Elan	1014	—	DIPs: 5C032, 5C060, 5C090 w/socket converter from outside source—PLCCs: 5C060, 5C090
	5-145	—	DIPs: 5C032, 5C060, 5C090 w/socket converter from outside source—PLCCs: 5C060, 5C090



Intel	iUP-PC iUP-200A/201A	GUIP Base (both prog.)	GUIP LOGICIID—DIPs: 85C060, 85C090, 5AC312, 5C060, 5C090 w/ADAPT24TO28—PLCCs: 85C060, 5AC312, 5C060 w/ADAPT40TO44—85C090 (PLCC), 5C090 (PLCC) GUIP 20D20J—85C220 (DIP & PLCC), 5C032 (DIP), 5C031 (DIP) GUIP 24D28J—85C224 (DIP & PLCC) GUIP 40D44J—5AC324 (DIP & PLCC) GUIP LOGIC-18—5C180 (PLCC) GUIP LOGIC-18G—5C180 (PGA)
Logical Dev.	ALLPRO	—	V2.1—DIPs: 85C220, 85C224, 85C060, 85C090, 85C22V10, PLD610, PLD910, PLD22V10, 85C508, 85C960, 5AC312, 5AC324, 5C031, 5C032, 5C060, 5C090 w/socket converters from outside source—PLCCs: 85C220, 85C224, 85C060, 85C090, 85C508, 5AC312, 5AC324, 5C060, 5C090, 5C180
Minato	1890A	—	5C031, 5C032, 5C060, 5C090, 5AC312, 85C060, 85C220, 85C508, 85C960
Oliver Adv. Eng.	OMNI-28	—	DIPs: 85C220, 5C031, 5C032, 85C224, 85C060, 5AC312, 5C060 w/OM-S-20 LCC—85C220 (PLCC) w/OM-S-24 LCC—85C224 (PLCC), 85C060 (PLCC), 5C060 (PLCC)
	OMNI-40	—	All DIPs above + DIPs: 85C090, 5AC324, 5C090, 5C121 w/OM-S-40 LCC—85C090 (PLCC), 5AC324 (PLCC), 5C090 (PLCC)
	OMNI-64	—	All DIPs above w/OM-S-68 LCC—5C180 (PLCC)
SMS GmbH	Sprint Plus	— w/Pod	V3.2i—DIPs: 5C031, 5C032, 5C060, 5C090, 5AC312 5C180 (PLCC)
Stag	ZL-30	—	DIPs: 5AC312, 5C031, 5C032, 5C060
	ZL30A	—	DIPs: 5AC312, 5C031, 5C032, 5C060
		w/30A640	5AC312 (PLCC), 5C060 (PLCC), 5C090 (DIP & PLCC)
	ZL33	—	DIPs: 5C031, 5C032, 5C060
	PPZ	—	DIPs: 5C031, 5C032, 5C060, 5C090
	System 3000	—	DIPs: 5AC312, 5C031, 5C032, 5C060, 5C090, 5C180
System General	SGUP-85A	—	V2.0—DIPs: 85C220, 85C224, 85C508, 85C960, 5AC312, 5C031, 5C032, 5C060, 5C090 w/adapters—PLCCs: 85C220, 85C224, 85C508, 85C960, 5AC312, 5C060, 5C090
	TURPRO-1	—	V1.1—85C220, 85C224, 85C508, 85C090, 5AC312, 5AC324, 5C031, 5C032, 5C060, 5C090, 5C180

## PLD PROGRAMMING VENDORS

The following table lists programming vendors and their products that support Intel PLDs.

Vendor Information	Product	Description
Advin Systems Inc. 1050 E. Duane Ave., Suite L Sunnyvale, CA 94086 (408) 984-8600	Sailor PAL/SA Sailor PAL/SB	Univ. 28 Pins + Expansion Modules Univ. 28 Pins + Expansion Modules Univ. 24-, 28-, 32-, or 40-Pin Programmers + Expansion Modules (PLCC)
BP Microsystems 10681 Haddington, # 190 Houston, TX 77043 (800) 225-2102 or (713) 461-9430	PLD-1100 PLD-1128	20/24 Pins + PLCC Socket Converters 28 Pins + PLCC Socket Converter
Bytek Corp. Instrument Systems Division 543 N. W. 77th St. Boca Raton, FL 33487 (800) 523-1565, (407) 994-3520 FAX (407) 994-3615	135H-U 145H-U	Universal MULTIPROGRAMMER® Logic Programmer
Data I/O Corp. 10525 Willows Road, N.E. P.O. Box 97046 Redmond, WA 98073-9746 (800) 247-5700 or (206) 881-6444 FAX (206) 882-1043	UNISITE 40/48 Model 2900 Model 29B Model 60A/H	Univ. 68 Pins—CHIPSITE for PLCC Univ. 40 Pins—DIP and PLCC Univ. 40 Pins—LogicPak for PLDs Univ. 28 Pins—H = Handler Version
Digelec Inc. 20144 Plummer St. Chatsworth, CA 91311 (800) 367-8750 or (818) 701-9677 FAX (818) 701-5040	Uniport	PLD Programmer + PLCC Adaptor
Elan Digital Systems 538 Valley Way Milpitas, CA 95035 (800) 541-3526 or (408) 946-8495 FAX (408) 946-0351	1014 5-145	Univ. + PLCC Socket Converters Programmer + PLCC Socket Converters
Intel Corporation (See Sales Office and Distributor Listings at Back of Handbook.)	iUP-PC iUP-200A/201A	Intel PC-Based + GUPI Adaptors Intel Standalone + GUPI Adaptors
Logical Devices, Inc. 1201 N.W. 65th Place Ft. Lauderdale, FL 33309 (800) 331-7766 or (305) 974-0967 FAX (305) 974-8531	ALLPRO  ALLPRO 88	Univ. 32 or 40 Pins + PLCC Socket Converters  Univ 88-Pin + PLCC Socket Converters

## PLD PROGRAMMING VENDORS (Continued)

The following table lists programming vendors and their products that support Intel PLDs.

Vendor Information	Product	Description
SMS GmbH Im Morgenthal 13 D-8994 HERGATZ Germany	Sprint Plus	Univ. 28 Pin + Expansion Pods
Stag Microsystems, Inc. 1600 Wyatt Drive, Suite 3 Santa Clara, CA 95054 (800) 227-8836 or (408) 988-1118 FAX (408) 988-1232	ZL-30 ZL-30A ZL33 PPZ	Logic 28 Pin + PLCC Adaptor Logic 28 Pin + PLCC Adaptor Gang 24 Pin Univ. 28 Pin + PLCC
System General Corp. 244 S. Hillview Dr. Milpitas, CA 95035 (408) 263-6667 FAX (408) 262-9227	SGUP-85A	Univ. 68 Pins

Univ. = Universal Programmer

## PLD SOFTWARE SUPPORT

Vendor	Product	Version/Devices Supported
Data I/O	ABEL	V4.2—85C220, 85C224, 85C060, 85C090, 85C508, 85C22V10, PLD610, PLD910, PLD22V10, 5AC312, 5AC324, 5C031, 5C032, 5C060, 5C090, 5C180
Intel Corp.	IPLDview-286	(Based on Workview V4.0)—85C220, 85C224, 85C060, 85C090, 85C508, 85C960, 5AC312, 5AC324, 5C031, 5C032, 5C060, 5C090, 5C180
ISDATA	Log/IC	V3.3—85C220, 85C224, 85C508, 5AC312, 5C031, 5C032, 5C060, 5C090, 5C180
Logic Modeling Corporation	Model Library	5AC312, 5AC324, 5C031, 5C032, 5C060, 5C090, 5C180, 85C22V10, 85C090, 85C060, PLD910, PLD610, PLD22V10, 85C220, 85C224
Logical Devices	CUPL	V4.2—85C220, 85C224, 85C060, 85C090, 85C508, 85C22V10, PLD610, PLD910, PLD22V10, 5AC312, 5AC324, 5C031, 5C032, 5C060, 5C090, 5C180
MINC Inc.	PLD Designer	V4.2—5C031, 5C032, 5C060, 5C090, 5C180, 85C220, 88C22V10, 85C508, PLD610, PLD910, PLD22V10, 5AC312, 85C224, 85C060, 85C090
OrCAD Systems	PLD Tools	V1.03—85C220, 85C224, 85C060, 85C090, 85C508, 5AC312, 5AC324, 5C031, 5C032, 5C060, 5C090, 5C180, 85C22V10, PLD610, PLD910, PLD22V10
Silicon West	Simulation Models	85C220, 85C224, 85C508, 5C032, 5C060, 5C090
Viewlogic, Inc.	Workview CAE	V4.1—85C220, 85C224, 85C060, 85C090, 85C508, 85C960, 5AC312, 5AC324, 5C031, 5C032, 5C060, 5C090, 5C180



The following table lists programming vendors and their products that support Intel PLDs.

Vendor Information	Product	Description
Data I/O Corp. 10525 Willows Road, N.E. P.O. Box 97046 Redmond, WA 98073-9746 (206) 881-6444	ABEL GATES PLDTest	Logic Compiler S/W Logic Synthesis S/W PLD Test S/W
Hewlett-Packard Company Customer Information Center (1-800) 752-0900	HP PLD Design System	Compiler/Synthesis S/W
Intel Corporation (See Sales Office and Distributor Listings at Back of Handbook.)	PLD Shell SCHEMA III-PLD iPLDview-286	Compiler S/W Schematic S/W Schematic/Simulation S/W
ISDATA GmbH Haid-und-New-Straße 7 D-7500 Karlsruhe West Germany	LOG/iC (Outside U.S. & Canada)	Compiler/Partitioner S/W
ISDATA, Inc. 800 Airport Rd. Monterey, CA 93940 (408) 373-3607	LOG/iC (U.S. & Canada)	Compiler/Partitioner S/W
Logical Devices, Inc. 1201 N.W. 65th Place Ft. Lauderdale, FL 33309 (800) 331-7766 or (305) 974-0967 FAX (305) 974-8531	CUPL	Compiler S/W
MINC, Inc. 6755 Earl Drive Colorado Springs, CO 80918 (719) 590-1155	PLDesigner	Compiler/Partitioner S/W
OrCAD Systems Corp. 3175 N.W. Alcock Dr. Hillsboro, OR 97124-7135 (503) 690-9881	OrCAD/SDT III OrCAD/VST	Schematic S/W Simulation S/W
Viewlogic, Inc. 313 Boston Post Road West Marlboro, MA 01752 (508) 480-0881	Intel PLD Kit	Intel PLD Design Library for Workview— Schematic/Simulation S/W



## PLDshell Plus Design Software

- **Easy-to-Use Design Environment for Programmable Logic Design Using Intel  $\mu$ PLDs and the FLEXlogic Family of FPGAs**
- **Compiles PALASM\* 2-Compatible Source Files into JEDEC Programming Files**
- **Functionally Simulates Designs Using Source File Simulation Syntax**
- **Easy-to-Learn Menu Interface with Extensive Help and On-Line Technical Support Information**
- **Editor and Programming Software Configurable to Designer's Preferred Tools**
- **Merges Multiple PDS Files into Single Device**
- **Estimates Design Fit Quickly to Narrow Target Device Choices**
- **Run Menu Configurable to Allow Use with Designer's Existing PLD Design Tools**
- **Utilities Disassemble/Convert Common PAL/GAL JEDEC Files into Source or JEDEC Files for Intel  $\mu$ PLDs**
- **Translates ADF/SMF Files into PDS Source Files**
- **Includes APT Programming Software for Use with Intel Programmers**
- **Software and Programming Hardware Available through Intel Authorized Sales Office/Distributor**
- **Standalone Software (No Programming Hardware) Available Free from Intel Literature**

### INTRODUCTION

PLDshell Plus design software provides an easy-to-use menu system for programmable logic design that allows you to invoke Intel's PLDasm compiler/simulation software and APT programming software, or your existing programmable logic compilers and programming software. Configure the menu system with the program and directory names of your existing programmable logic design tools and you are ready to run.

PLDshell Plus software includes Intel's PLDasm logic compiler/simulator software. PLDasm software compiles PALASM\* 2-compatible source files to produce JEDEC files for Intel  $\mu$ PLDs (Microcomputer Programmable Logic Devices) and FLEXlogic family of FPGAs. Simulation syntax supports functional simulation and generation of test vectors for the JEDEC file. PLDasm software allows you to use a familiar design language to evaluate the architecture of Intel  $\mu$ PLDs and FPGAs and to implement new designs.

\*PALASM and PAL are registered trademarks of Advanced Micro Devices, Inc.  
\*GAL is a registered trademark of Lattice Semiconductor, Inc.

## PLDshell Plus OVERVIEW

The PLDshell Plus Main Menu is arranged to follow the typical PLD design flow: Edit, Compile/Simulate, View, and Program. PLDshell Plus menu options allow you to:

**Edit**—Edit PLD source files (or any other text file) using your preferred ASCII text editor. You can configure the edit menu to invoke whatever text editor you have installed in your system via the **Change Editor** button.

**Compile/Sim**—Compile and/or simulate PLDasm source files to create JEDEC files for Intel  $\mu$ PLDs using Intel's PLDasm software. You can define compile/simulate options such as logic minimization, DeMorgan's inversion, pin assignment algorithm options, and event threshold for asynchronous events (during simulation).

**View**—View source, error, report, or simulation files to validate your design or quickly locate design or fitting problems. When viewing an error file, you can display on-line error message help information. Simulation results can be viewed in table or wave form. (Waveforms can be viewed on Hercules, EGA, or VGA monitors.)

**Program**—Invoke Intel's APT programming software. You can change the program menu to your preferred programming software via the **Change Programming S/W** button.

**Run**—Run up to 24 user-defined programs including other PLD development tools. The menu is user-defined, with each menu option including default command line options and working directories. You can also run any program via the **Run Other** submenu.

**Utilities**—Provides several utility functions, including:

**Disassemble**, allows you to disassemble JEDEC files for common PAL/GAL devices into PLDasm source files for Intel  $\mu$ PLDs (JEDEC file to PDS file).

**Convert**, performs a full conversion (common PAL/GAL JEDEC file to Intel  $\mu$ PLD JEDEC file).

**Translate**, translates ADF/SMF files developed for Intel's iPLS II software into PDS files for use with PLDasm software.

**Merge**, allows you to combine multiple PDS files into a single Intel PLD or FPGA.

**Change Directory, List Directory, or Invoke DOS Shell.**

**Modify Options** to change the text editor, programming software, printer port, hot keys, and other options.

**Databook**—View datasheet briefs on Intel  $\mu$ PLDs and technical notes on using the software/devices, device order codes, and other pertinent technical information.

## PLDasm COMPILATION/SIMULATION

PLDasm software compiles PALASM 2-compatible source files to produce JEDEC files for Intel  $\mu$ PLDs. The typical design process is to create the source file, compile/simulate the design to create a JEDEC file, and program devices. Error, report, and waveform files are viewed throughout the cycle. The edit/compile/simulate/view process is repeated until a design is working as desired. Devices are programmed at the end of the cycle. PLDshell Plus software also offers the ability to edit/simulation/view without "fitting". This allows you to concentrate on logic design first, then fit the working design into a device.

PLDasm software offers the following features:

- Preserves your investment in learning a PLD design language and in developing source files by compiling an industry-standard language.
- Implements designs using Boolean equations, State Machine syntax, or Truth Tables (*Truth Table design is a PLDasm superset feature*).
- Functionally simulates designs.
- Estimates design fit quickly based on basic resource criteria.
- Maps designs into device resources and performs basic logic minimization.
- Generates JEDEC programming files; these files include programming test vectors based on simulation output.

You can compile .PDS files for common PAL/GAL devices using PLDasm software. PAL/GAL designs are transparently converted into JEDEC files for the appropriate Intel  $\mu$ PLD.

## DESIGN MERGE

PLDshell Plus can merge multiple PDS design files into any Intel programmable logic device, including the Intel FLEXlogic family of FPGAs. The Merge function makes it easy for designers to consolidate multiple PLDs into a single, high-performance FPGA or PLD.

## JEDEC DISASSEMBLY

PLDshell Plus provides the ability to disassemble existing JEDEC files for Intel  $\mu$ PLDs, as well as for common 20-pin and 24-pin PALs and GALs into PLDasm source files. JEDEC disassembly allows you to reconstruct source files for existing designs where the original source files have been lost, or to generate source files from existing designs to be modified for new designs. JEDEC disassembly is available via the **Utilities—Disassemble** menu selections. Note that the source file output during the disassembly process using PAL/GAL JEDEC files is for the respective Intel  $\mu$ PLD.

## JEDEC CONVERSION

PLDshell Plus provides the ability to convert existing JEDEC files for common PALs/GALs into JEDEC files for Intel  $\mu$ PLDs. A PLDasm source file is automatically generated during the conversion process. Conversion guarantees that the target Intel  $\mu$ PLD is functionally the same as the original design. JEDEC conversion is available via the **Utilities—Convert** menu selections.

## TRANSLATION

PLDshell Plus provides the ability to translate existing ADFs (Advanced Design Files) or SMFs (State Machine Files) into PDS files for use with PLDshell Plus. This provides an automated path for users of Intel's iPLS II software to move to PLDshell Plus and make use of its more powerful simulation features and user interface. Translations is available via the **Utilities—Translate** menu selections.

## SYSTEM REQUIREMENTS

PLDshell Plus is designed to work in systems configured as follows:

- Intel 386-based PC with 2 Mbytes of extended RAM
- MS-DOS V3.1 or later
- High-density (1.44 Mbytes) diskette drive; call Intel's EPLD Hot Line or your Intel field sales representative for other disk formats.
- Hard disk with approximately 5 Mbytes of space (4 Mbytes for installation; up to 1 Mbyte for working files while running).
- VGA monitor required for waveform viewing.

4

## ORDERING INFORMATION

PLDshell Plus software (no programming hardware) is available from Intel Literature or your local Intel salesperson. (Quantity limited per order.)

DSPLUSKIT includes PLDshell Plus software and the iUP-PC programming hardware. Available through Intel-authorized sales offices.

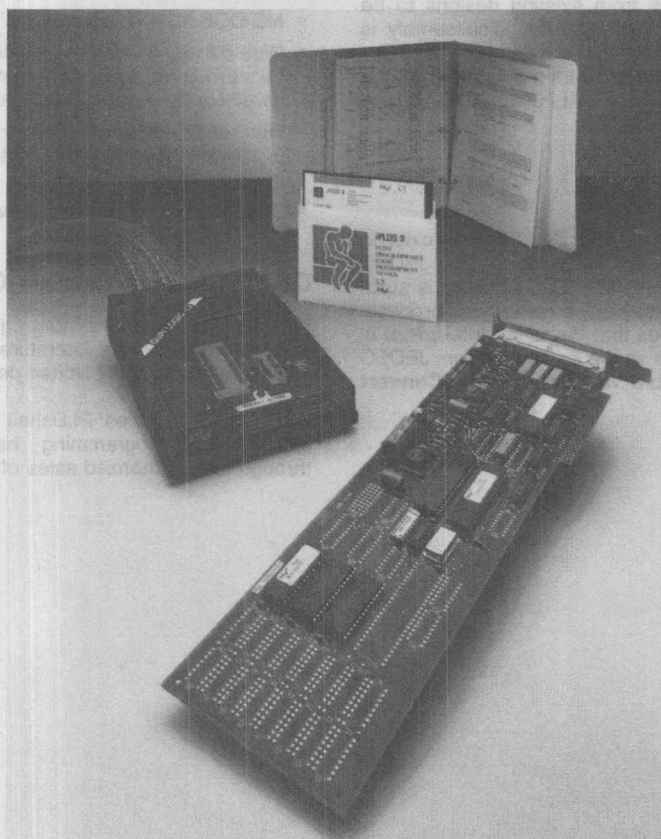




## iUP-PC UNIVERSAL PROGRAMMER FOR THE PERSONAL COMPUTER

- Personal Computer Version of the iUP-200A/201A Universal Programmers
- Runs on an IBM PC/AT\*, PC/XT\* or True Compatible
- GUPI Adaptors and Personality Modules Provide Support for Numerous Device Families
- Utilizes the Intelligent Programming and Quick-Pulse Programming™ Algorithms
- Extremely Versatile—Programs Intel EPROMs, PLDs, Peripherals, and MicroControllers

The Intel Universal Programmer for the Personal Computer, iUP-PC, provides a high-performance programming solution from a PC host. Through plug-in adaptors for the Generic Universal Programmer Interface (iUP-GUPI), the iUP-PC supports Intel EPLDs and most other Intel programmable devices.



This product is manufactured by Intel Puerto Rico, Inc.

**NOTE:**

GUPI Adaptor NOT included.

\*IBM PC/AT and PC/XT are registered trademarks of International Business Machines Corporation. Quick-Pulse Programming is a trademark of Intel Corporation.

290130-1

## FUNCTIONAL DESCRIPTION

The iUP-PC provides a fast, versatile and reliable programming solution from a Personal Computer host. Downloading to a stand-alone programmer or moving from one workstation to another is no longer required. With the iUP-PC, the designer may do his development and programming on one workstation. Through the Generic Universal Programmer Interface (iUP-GUPI), the iUP-PC is made extremely versatile. With the iUP-GUPI the designer may program supported EPROMs, Microcontrollers, Peripherals and PLDs with the mere change of a plug-in adaptor. As all of the programming signals are generated at the GUPI base, extremely reliable waveforms reach the device.

## COMPONENTS

The iUP-PC programming system consists of five components:

**PCPP**—The Personal Computer Personal Programmer (PCPP) is an IBM PC/XT form factor expansion card which fits into an IBM PC/XT, PC/AT or true compatible.

**Interconnect Cable**—A 50-lead ribbon cable connects the PCPP to the iUP-GUPI.

**iUP-GUPI**—The Intel Universal Programmer—Generic Universal Programmer Interface (iUP-GUPI) is the programming base which holds the device adaptors.

**GUPI Adaptors\***—The GUPI Adaptors plug in to the iUP-GUPI base. They carry the sockets and hardware for a particular device family.

**iPPS**—The Intel PROM Programmer Software (iPPS) runs on a personal computer under DOS and controls the PCPP/host communication.

### \*NOTE:

Though the iUP-GUPI base is included in the iUP-PC package, the GUPI Adaptors are NOT included. The desired adaptors must be ordered separately.

## PCPP CARD

The PCPP is a co-processor board. Communication between the host and the PCPP may be controlled by iPPS or APT (Advanced Programming Tool). Version 2.3 or greater of iPPS is required for running the iUP-PC on a personal computer. APT is the programming software product included in PLDshell Plus Design Software. The PCPP is capable of driving the iUP-GUPI and FAST27/K modules.

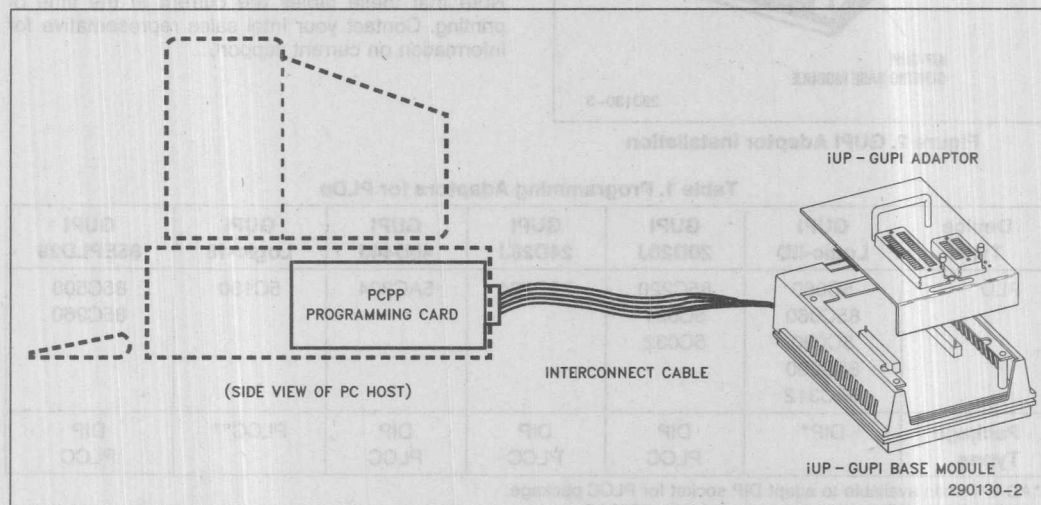


Figure 1. The Intel Universal Programmer for the Personal Computer (iUP-PC)

The iUP-GUPI is a generic base module that enables the iUP-PC system to accept low-cost plug-in adaptors. These adaptors configure the system to support a wide variety of programmable devices—EPROMs, microcontrollers, and PLDs—as well as different device package types (either directly or via socket converters).

The iUP-GUPI module connects to the PCPP card via a ribbon cable. An opening in the top of the iUP-GUPI provides easy plug-in installation of the GUPI adaptors (refer to Figure 2).

The iUP-GUPI offers the programming performance of earlier Intel personality modules, with the fastest Intel programming algorithms for each device type.

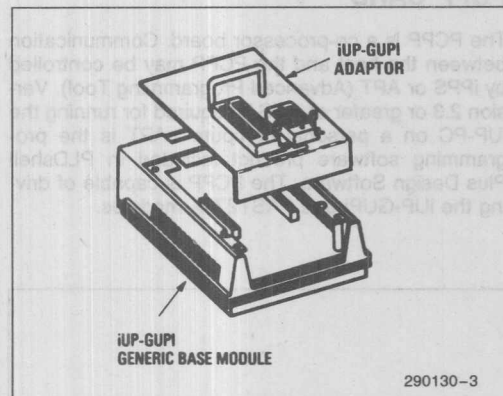


Figure 2. GUPI Adaptor Installation

The iUP-GUPI adaptors provide the final link of the iUP-PC programming system. The adaptors provide the proper sockets and characteristic information for devices or families of devices.

Intel's megabit EPROMs are also supported with GUPI adaptors. Adaptors are available for the 1-, 2-, and 4-Megabit EPROMs in both byte- and word-wide architectures.

The GUPI27011 adaptor, which supported the page-mode 27011 is now replaced by GUPI EPROM28, which also adds support for the CMOS version (27C011). Devices supported in the past by the iUP-FAST 27/K Personality Module are now supported by the GUPI EPROM28 adaptor and the GUPI base module. The GUPI EPROM28/iUP-GUPI base combination does not support the following obsolete devices: 2764, 27128, 27916, 2817A, and 87C256. The GUPI EPROM28 adaptor corrects/updates programming algorithms from the iUP-FAST 27/K and upgrade kits.

Table 1 lists PLD GUPI adaptors and the devices they support. Table 2 lists EPROM adaptors and the devices they support. Table 3 lists programming support for Microcontrollers. Table 4 lists the programming adaptors for automotive Microcontrollers. Note that these tables are current at the time of printing. Contact your Intel sales representative for information on current support.

Table 1. Programming Adaptors for PLDs

Device Type	GUPI Logic-IID	GUPI 20D20J	GUPI 24D28J	GUPI 40D44J	GUPI Logic-18	GUPI 85EPLD28
PLD	5C060 85C060 5C090 85C090 5AC312	85C220 5C031 5C032	85C224	5AC324	5C180	85C508 85C960
Package Types	DIP*	DIP PLCC	DIP PLCC	DIP PLCC	PLCC**	DIP PLCC

\*ADAPT units available to adapt DIP socket for PLCC package.

\*\*GUPI LOGIC-18G is available to support the 5C180G.

Table 2. Programming Adaptors for EPROMs

Device Type	GUPI 27010	GUPI 27210	GUPI 27960	GUPI EPROM28
EPROM	27010	27210	27960CX	2764A
	27C010	27C210	27960KX	27C64
	27C010A*	27C220		87C64
	27C100	27C240		27128A
	27C020	27C400**		27C128
	27C040	27C202		27256
		87C75PF		27C256
				68C257
				87C257
				27512
				27C512
				27513
				27C513
				27011
				27C011
Package Types	DIP	DIP	CJCC PLCC	DIP

\*Programming algorithm available on request.

\*\*Supported with 27C240 to 27C400 pin conversion adaptor and ID override.

Table 3. Programming Adaptors for Microcontrollers

Device Type	GUPI 8742	GUPI 8796	GUPI MCS-96LCC
Peripheral	8741AH 8742AH		
Microcontroller		8796BH 8794BH 8795BH 8797BH	8796JC 8797BH 87C196KB
Package Types	DIP	PGA DIP	LCC

Table 4. Programming Adaptors for Automotive Microcontrollers

Device Type	GUPIMCS96	GUPIC196	GUPIC196KX	GUPI196KJ	GUPI87C51GBPLCC	GUPIMCS51FX
Microcontroller	8796BH	87C194	87C196KR	87C196KJ	87C51GB	87C51
	879XJC	87C198	87C196JR			87C51FA
	8797JF	87C196KB	87C196KQ			87C51FB
		87C196KC	87C196JQ			87C51FC
		87C196KD				87C54
						87C58
Package Types	PLCC	PLCC PLCC	PLCC PLCC	PLCC	PLCC	PLCC



## IPPS SOFTWARE

iPPS software, included with the iUP-PC, performs the following functions to make programming quick and easy:

- Reads EPROMs, EPLDs and microcontrollers.
- Programs devices from a preprogrammed master or from a file.
- Verifies devices data with buffer data.
- Prints devices buffer, or device file contents to system printer.
- Performs interactive formatting operations such as interleaving, nibble swapping, bit reversal, and block moves.
- Programs multiple devices from source file, prompting user to insert new devices.
- Buffer contents can be edited.

With iPPS software, the user can load programs from system memory or directly from a disk file. Access to the disk lets the user create and manipulate data in a virtual buffer. This block of data can be formatted for use with byte- or word-wide devices.

iPPS software supports data manipulation in the following Intel formats: 8080 hexadecimal ASCII, 8080 absolute object, 8086 hexadecimal ASCII, 8086 absolute object, 80286 absolute object, and 80386 bootloadable object. Addresses and data can be displayed in binary, octal, decimal, or hexadecimal. The user can easily change default data formats as well as number bases. Object module formats from some 80386 and later compilers and assemblers are not compatible with the iPPS software.

## IUP-PC SPECIFICATIONS

### HOST SYSTEM

iPPS runs on an IBM PC/XT, PC/AT or other true compatible with a DOS operating system. The PCPP requires one full-sized card slot inside the PC.

### OPERATING ENVIRONMENT

#### Electrical Characteristics

##### PCPP:

##### Worst-Case Power Consumption at IBM PC I/O Channel

Supply Voltage	Voltage Variance	Max. Current Drain
+ 5V	+ 5%, - 4%	1.898 A
- 12V	+ 10%, - 9%	102.9 mA
+ 12V	+ 5%, - 4%	530 mA

#### Physical Characteristics

##### PCPP:

Length: 13.3 inches (33.9 cm)  
Height: 3.9 inches (10.0 cm)

##### Interconnect Cable:

50 lead ribbon cable  
Length: 3.0 feet (91.4 cm)  
Width: 2.43 inches (5.5 cm)

##### IUP-GUPI:

Length: 7.0 inches (17.8 cm)  
Width: 5.5 inches (1.4 cm)  
Height: 1.6 inches (4.1 cm)

## Environmental Characteristics

### Environmental Class: B

#### Temperature:

Reading	10°C to 40°C
Programming	25°C ± 5°C
Operating	10°C to 40°C
Non-Operating	-40°C to 70°C

#### Relative Humidity:

Operating	85% Maximum
Non-Operating	95% Maximum

## DOCUMENTATION

168161—PCPP User's Guide

166428—iUP-GUPI Module User's Guide

## ORDERING INFORMATION

### Order Code

### Product Description

iUPPC

Universal Programmer for the Personal Computer: PCPP Programming Card, 50-Lead Interconnect Cable, iUP-GUPI, iPPS, PCPP User's Guide

ADAPT24TO28

28-Pin PLCC Socket Adaptor for GUPI LOGIC-IID

ADAPT40TO44

44-Pin PLCC Socket Adaptor for GUPI LOGIC-IID

piUPGUPI

Generic Universal Programmer Interface (Base)

GUPI LOGICIID

GUPI Logic Adaptor

GUPI40D44J

GUPI Logic Adaptor

GUPI85EPLD28

GUPI Logic Adaptor

GUPI 20D20J

GUPI Logic Adaptor

GUPI 24D28J

GUPI Logic Adaptor

GUPI LOGIC18

GUPI Logic Adaptor

GUPI LOGIC18G

GUPI Logic Adaptor for 5C180 PGA

GUPI27010

iUP-GUPI EPROM Adaptor

GUPI27210

iUP-GUPI EPROM Adaptor

GUPI27960

iUP-GUPI EPROM Adaptor

GUPI EPROM28

iUP-GUPI EPROM Adaptor

GUPI8742

iUP-GUPI Peripheral Adaptor

GUPI8796

iUP-GUPI Microcontroller Adaptor

GUPI MCS-96LCC

iUP-GUPI Microcontroller Adaptor

piUPFAST 27K

(Order GUPI EPROM28 and piUPGUPI)

iUPADAPT28TO32

32-Pin PLCC Socket Adaptor for 28-Pin DIP Sockets

## FROM PROGRAMMERS

- Provides Programming Support for Intel EPROMs, PLDs, Microcontrollers, and Peripherals
- PROM Programming Software (iPPS) Makes Programming Easy with IBM PC/XT\*, PC/AT\*, and PC Compatibles
- Supports Personality Modules and GUPI Base W/Adaptors
- iUP-200A Provides On-Line Operation with a Built-In Serial RS232 Interface and Software for a PC Environment
- iUP-201A Provides Same On-Line Performance and Adds Keyboard and Display for Stand-Alone Use
- iUP-201A Stand-Alone Capability Includes Device Previewing, Editing, Duplication, and Download from any Source Over RS232C Port

The iUP-200A and iUP-201A Intel universal programmers program and verify data in Intel programmable devices. The iUP-200A and iUP-201A provide on-line programming and verification using Intel's PROM programming software (iPPS). The iUP-201A supports off-line, stand-alone program editing, duplication, and memory locking.



These products manufactured by Intel Puerto Rico, Inc.

210319-1

\*IBM PC/XT and PC/AT are registered trademarks of International Business Machines Corporation.

## FUNCTIONAL DESCRIPTION

The iUP-200A universal programmer operates in on-line mode. The iUP-201A universal programmer operates in both on-line and off-line modes.

### On-Line System Hardware

The iUP-200A and iUP-201A universal programmers are free-standing units that, when connected to a host computer with at least 64K bytes of memory, provide on-line programming and verification of Intel programmable devices. In addition, the iUP can read the contents of the ROM versions of supported devices.

The iUP communicates with the host through a standard RS232C serial data link. Each iUP contains a CPU, selectable power supply, static RAM, programmable timer, interface for personality modules, RS232C interface for the host system, control firmware in EPROM, and a connector to the GUPI (Generic Universal Programmer Interface) base. The iUP-201A also has a keyboard and display.

A personality module or GUPI Adaptor adapts the iUP to a family of devices; it contains all the hardware and software necessary to program either a family of devices or a single Intel device. The user inserts the personality module/GUPI base into the universal programmer front panel. An iUPDL Upgrade Kit supports down-loadable algorithm files.

### On-Line System Software

The iUP-200A and iUP201A includes Intel's PROM Programming software iPPS. iPPS software provides user control through an easy-to-use interactive interface. iPPS software performs the following functions to make EPROM programming quick and easy:

- Reads devices
- Programs devices from a pre-programmed master or from a file
- Verifies device data with buffer data
- Locks device memory from unauthorized access (on devices which support this feature)
- Prints device contents to a system printer

- Performs interactive formatting operations such as interleaving, nibble swapping, bit reversal, and block moves
- Programs multiple devices from the source file, prompting the user to insert new devices
- Buffer contents can be edited

All iPPS commands, as well as program address and data information, are entered through the host system ASCII keyboard and displayed on the system CRT.

iPPS software supports data manipulation in the following Intel formats: 8080 hexadecimal ASCII, 8080 absolute object, 8086 hexadecimal ASCII, 8086 absolute object, 80286 absolute object, and 80386 boot-loadable object. Addresses and data can be displayed in binary, octal, decimal, or hexadecimal. The user can easily change default data formats as well as number bases. iPPS can also access disk files.

Object module formats from some 80386 and later compilers and assemblers are not compatible with the iPPS software.

For programming Intel PLDs, the iUP-200A/201A can be controlled by Intel's Advanced Programming Tool (APT). APT programs PLDs from JEDEC files produced by Intel's logic compiler. (iPPS can also program PLDs, but only from pre-programmed device masters.)

### Off-Line System

The iUP-201A has all the on-line features of the iUP-200A plus off-line editing, device duplication, program verification, and locking of device memory independent of the host system. The iUP-201A also accepts Intel hexadecimal programs developed on non-Intel development systems. Just a few key-strokes download the program into the iUP RAM for editing and loading into a device.

Off-line commands are entered via a 16-character keypad. A 24-character display shows programmer status.



## PERSONALITY MODULES

For some devices, a personality module is the interface between the iUP universal programmer and a selected device. Personality modules contain all the hardware and firmware for reading and programming a family of Intel devices. Table 1 lists the devices supported by the different modules.

## GUPI ADAPTORS

For many devices, the GUPI base module and interchangeable GUPI Adaptors provide the interface between the programmer and the device being programmed (see Figure 1). The GUPI module is a base module that interfaces to the iUP-200A/201A and GUPI Adaptors. GUPI Adaptors tailor the GUPI module base signals to a family of devices or an individual device.

The GUPI 27011 adaptor, which supported the page-mode 27011, is now replaced by the GUPI EPROM28, which also adds support for the CMOS version (27C011). Devices supported in the past by the iUP-FAST27/K Personality Module are now supported by the GUPI EPROM28 adaptor and the GUPI base module. The GUPI EPROM28/iUP-GUPI base combination does not support the following obsolete devices: 2764, 27128, 27916, 2817A, 87C256. The GUPI EPROM28 adaptor corrects/updates programming algorithms from the iUP-FAST27/K and upgrade kits.

Tables 2 through 5 show which Adaptors support which devices. Note that these tables are current at the time of printing. Contact your Intel sales representative for information on current support.

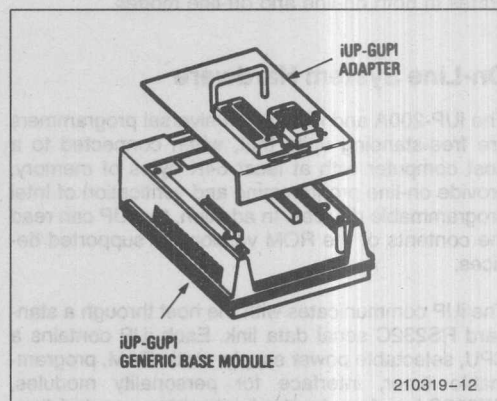


Figure 1. GUPI Base and Adaptor

Table 1. iUP Personality Programming Modules

Device Type	Fast 27/K Module	F27/128 Module	F87/44A Module	F87/51A Module
EPROM	Obsoleted (See GUPI EPROM28)	2716 2732 2732A 2764 27128		
Microcontroller			8041A 8042 8044AH 8741H 8742 8744H 8755A	8748 8748H 8749H 8751 8751H 8048 8048H 8049 8049H 8050H 8051

\*Quick-Pulse Programming™ algorithm

Table 2. Programming Adaptors for Memories

Device Type	GUPI 27010	GUPI 27210	GUPI 27960	GUPI EPROM28
EPROM	27010 27C010 27C010A* 27C100 27C020 27C040	27210 27C210 27C220 27C240 27C400** 27C202 87C75PF	27960CX 27960KX	2764A 27C64 87C64 27128A 27C128 27256 27C256 68C257 87C257 27512 27C512 27513 27C513 27011 27C011
Package Types	DIP	DIP	CJCC PLCC	DIP

**NOTE:**

\*Programming algorithm available on request.

\*\*Supported with 27C240 to 27C400 pin conversion adaptor and ID override.

Device Type	GUI Logic-IIID	GUI 20D20J	GUI 24D28J	GUI 40D44J	GUI Logic-18	GUI 85EPLD28
PLD	iPLD610 5C060 85C060 iPLD910 5C090 85C090 5AC312	85C220 5C032 5C031	85C224	5AC324	5C180	85C508 85C960
Package Types	DIP*	DIP PLCC	DIP PLCC	DIP PLCC	PLCC**	DIP PLCC

\*ADAPT units available to adapt DIP socket for PLCC package.

\*\*GUI LOGIC-18PGA is available to support the 5C180.

**Table 4. Programming Adaptors for Microcontrollers**

Device Type	GUI 8742	GUI 8796	GUI MCS-96LCC
Peripheral	8741AH 8742AH		
Microcontroller		8794BH 8795BH 8796BH 8797BH	8796JC 8797BH 87C196KB
Package Types	DIP	PGA DIP	LCC

**Table 5. Programming Adaptors for Automotive Microcontrollers**

Device Type	GUPIMCS96	GUPIC196	GUPIC196KX	GUPI196KJ	GUPI87C51GBPLCC	GUPIMCS51FX
Microcontroller	8796BH 879XJC 8797JF	87C194 87C198 87C196KB 87C196KC 87C196KD	87C196KR 87C196JR 87C196KQ 87C196JQ	87C196KJ	87C51GB	87C51 87C51FA 87C51FB 87C51FC 87C54 87C58
Package Types	PLCC	PLCC	PLCC	PLCC	PLCC	PLCC

## IUP-200A/201A SPECIFICATIONS

### Physical Characteristics

Depth: 15 inches (38.1 cm)  
Width: 15 inches (38.1 cm)  
Height: 6 inches (15.2 cm)  
Weight: 15 pounds (6.9 kg)

### Electrical Characteristics

Selectable 100, 120, 200, or 240 Vac  $\pm 10\%$ ; 50-60 Hz  
Maximum power consumption—80 watts

### Environmental Characteristics

Reading Temperature: 10°C to 40°C  
Programming Temperature: 25°C  $\pm 5^\circ$   
Operating Humidity: 10% to 85% relative humidity

### Reference Material

- 166041-001—*iUP-200A/201A Universal Programmer User's Guide.*
- 166042-001—*Getting Started with the iUP-200A/201A (For ISIS/iNDX Users).*
- 166043-001—*Getting Started with the iUP-200A/201A (For DOS Users).*
- 164853 — *iUP-200A/201A Universal Programmer Pocket Reference.*

## ORDERING INFORMATION

Product Order Code	Description
PIUP200A 216D	On-line PROM programmer with iPPS rel 2.0 for PC/DOS, and cable for PC or XT
PIUP200A 217D	On-line PROM programmer with iPPS rel 2.0 for PC/DOS, and cable for AT
PIUP201A 216D	Off-line and on-line PROM programmer with iPPS rel 2.0 for PC/DOS, and cable for PC or XT
PIUP201A 217D	Off-line and on-line PROM programmer with iPPS rel 2.0 for PC/DOS, and cable for AT
PIUP200/201 U1* Upgrade Kit	Upgrades an iUP-200/201 to an iUP-200A/201A
IUPDL	Download Support Kit for iUP-200A/201A upgrades programmer to support adaptors that use disk-based software programming (.DSS) files.

\*Most personality modules can be used only with an iUP-200A/201A or an iUP-200/iUP201 upgraded to an A with the PIUP-200/201 U1 upgrade kit.

Product Order Code	Description
PIUP-GUPI	Generic Universal Programmer Interface (Base)

### Software Sold Separately

Product Order Code	Description
P216D	PROM programming software rel 2.0 for PC/DOS with cable for PC or PC/XT
P217D	PROM programming software rel 2.0 for PC/DOS with cable for PC/AT



# Personalizing PLD design work

By JAY STURGES  
SENIOR SOFTWARE ENGINEER  
INTEL CORP.  
FOLSOM, CALIF.



Recent years have brought a spate of advances in PLD-development tools and their underlying technology, including the transition from device-dependent, restricted PLD design languages to device-independent, Boolean equation and

finite state-machine design languages. An associated advancement is the integration of standard PLD development systems with major CAE vendor design environments. However, with all these improvements, there is still no intuitive PLD design tool set where a designer can achieve true universal PLD device and vendor support.

With hundreds of PLD architectures, made by dozens of IC vendors, it's not surprising that no single tool provides all the engineer's support. By the same token, a designer can't find a PLD design tool that can cover his various PLD design method and language requirements. Thus, engineers are forced to use multiple PLD tools to meet their needs and have additional learning requirements.

Currently, there are three distinct types of PLD-design tools—manufacturer-specific, universal and comprehensive. Each tool in these three categories offers the engineer some solutions to the development cycle. However, each has problems that lead the engineer to use more than one PLD-design tool.

Manufacturer-specific tools provide support for early development of new PLD architectures (before universal tools can bring up their support). By providing PLD tools, makers ensure that system-design engineers will have immediate development capability for their most-advanced PLD devices. Some manufacturers will

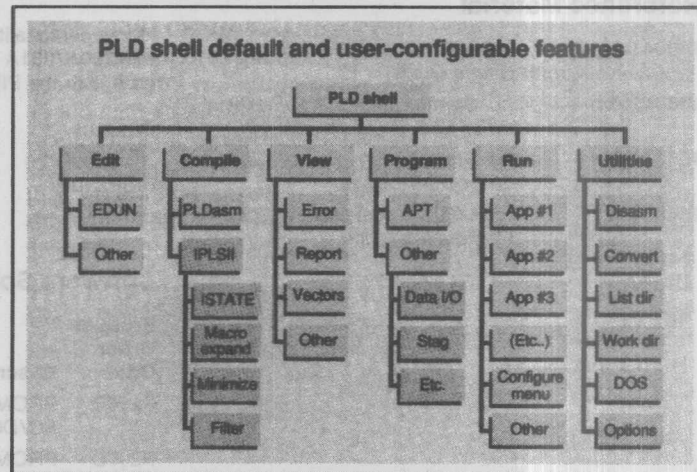
even make this capability available prior to the availability of new PLD silicon.

In addition, manufacturer-specific tools will generally provide highly optimized support for their new PLD architectures. By using proprietary fitting, allocation and simulation technologies, PLD makers can optimize the design tools to best use their family of devices.

The downside to using manufacturer-specific tools is that they generally don't support the architectures of other PLD makers. They also incorporate new functionality (requiring additional learning) so that advanced device support can be optimized. As an example, optimal support for a complex device, such as Intel's 5AC312, requires the ability to "borrow" groups of

Universal PLD tool vendors provide tools offering a level of PLD device and vendor independence. Universal vendors promote device-independent systems, sometimes including languages with higher levels of abstraction and partitioning algorithms. Their support includes as large a list of PLD manufacturers as possible. This gives the system design engineer the ability to select from a large list of PLD devices and vendors. Yet, trying to support a large list of architecturally different PLDs means that support for particular devices, including the most advanced ones, is commonly sacrificed.

Among the most comprehensive PLD tools are those supplied by major CAE



logic in the device and "allocate" them as the user targets his design in the device.

The company's manufacturer-specific tool does this using software algorithms but requires the user to leave pin definitions unassigned. Many users will not take advantage of this capability, because the other tools they employ require pin assignment prior to use. So, to use the tool, a change in PLD design method as compared with other PLD design tools is required.

vendors, who provide an entire design methodology integrating universal and manufacturer-specific PLD tools, as well as other semicustom capabilities. They try to solve design problems by providing a complete electronic design automation (EDA) methodology. Typically referred to as frameworks, these solutions are supported by alliances between the CAE vendors and manufacturer-specific and/or universal PLD tool vendors.

These alliances produce interfaces allowing the PLD tools to be integrated into the EDA methodology. But, because each alliance requires much development work, these alliances are generally limited to a small number of vendors. An engineer can't always get the tool set he requires within the framework. Furthermore, the costs of using these frameworks is often very high in both dollar outlays and engineer training.

For each of the three PLD-tool types, shortcomings exist for providing universal, low-cost support for the design of PLDs.

This introduces an idea for a new class of PLD tools: a PLD micro-framework. Such a tool should be built around the theme of letting the engineer "have it his way." It should provide a workbench for PLD development, providing design engineers with a way to integrate different PLD tools, methods and languages. It should be structured around an intuitive user interface.

The PLD micro-framework should provide a generic environment for editing, compiling, viewing, simulating, programming and even disassembling ID. These are the steps needed to generate a PLD design, independent of the PLD device or vendor selected. Within each of these steps, the engineer should be able to configure the programs he runs to his individual needs. There are a few requirements of a PLD micro-framework software design package:

- **Consistent user interface:** The tool should try to minimize the user's learning curve for incorporating new PLD capabilities by presenting users with a consistent, intuitive environment for PLD design. By entering various stages of PLD design via the framework, the user should be constantly aware of where he is in the design cycle and how he can move rapidly through the cycle. The PLD framework should act as a dispatcher, administering execution of the PLD design.

- **Preserve existing user-knowledge investment:** The PLD design cycle incorporates a series of separate software-design programs. A user will use a text editor to create a design file (e.g., Boolean expres-

sion, state-machine expression or net-list file). He then will proceed to a compiler, which will take the design description file created in the editor and translate it into a target device (e.g., minimize Boolean equations, produce Jedec programming files). After completing the design, the engineer will program the file into a target device.

The user may use different vendors for each of the programs. As an example, an AEDIT text editor, Intel PLD compiler (iPLS), and Data I/O programmer (e.g., Unisite) may be chosen. For each of these systems, an investment has been made in training. The framework should allow these investments to be preserved. This should be made possible by enabling the user to configure the framework for editing, compiling and programming software.

- **Allow integration of multiple-PLD compilers:** There's a constant stream of new PLD design tools (manufacturer-specific) to support new PLD devices. These enable fast time-to-market for support of new PLD products. These also can offer more optimum device utilization than can universal tools. The framework should provide an environment for running many different PLD compiler programs.

Users should be able to quickly add and inventory PLD tools from both manufacturer-specific and universal vendors. This would result in the ability of design engineers to move easily between different types of design tools to incorporate new PLD devices and better optimize their individual PLD designs.

- **Import and upgrade the PALasm syntax:** A large percentage of PLD designs use PALasm or a PALasm-like syntax. The PLD framework tool should incorporate a compiler that can import PALasm syntax (which assumes PALs as target devices) and, in fact, upgrade the language to allow for PLD-independent design. By migrating PALasm to this device-independent syntax, designs could be easily migrated between ID.

Users should be able to use the new language syntax to store device-independent design modules and use them at various levels of PLD integration in future

designs. Little learning would be required, since users are already familiar with PALasm. In addition, since most manufacturer-specific tools have the ability to import PALasm syntax, mobility would exist between these tools.

- **Additional PLD design utilities:** The framework should include the ability to disassemble Jedec files into source, view files quickly and move easily within the DOS environment. The disassemble utility would allow engineers to identify functionality of ID without source files readily accessible. By reading a device's Jedec pattern (for example, from a programmer), the designer can disassemble the Jedec into a source file that includes the Boolean representation of functionality.

Intel is now providing a tool aimed at filling the needs outlined above. The company's PLDshell program provides a new capability in PLD tools. It provides the micro-framework outlined above for PLD design. It includes a user interface tailored specifically for the PLD design engineer. Within this interface, the user can customize editors, compilers and programming environments to preserve past investment in learning (see figure).

Additional capability also has been provided specifically for PLD design. A new PLDasm compiler is included that allows new capabilities in a very familiar format. The user can upgrade designs initially targeted for PALs to use the advanced capabilities available in newer PLD devices. PALasm files can also be imported. Moreover, users can recover design information by disassembling existing Jedec files.



Since 1988, Jay Sturges has worked as a senior software engineer for Intel's programmable logic focus group. He has been involved in CAD and CAE for the past 10 years. Prior to Intel, he helped create SuperCads, a small development company in northern California.









## APPENDIX

### PLD Programming Support

Vendor	Product	Module	Adaptor/Device/Version/Etc.
Advin	Sailor PAL/SA	—	DIPs: 5C031, 5C032, 5C060
	Sailor PAL/SB	— w/EM-900 w/EM-1800 w/PLCC Adap.	(same as above) DIP: 5C090 PLCC: 5C180 PLCCs: 5C060, 5C090
BP Micro	PLD-1100	—	V1.46—DIPs: 85C220, 85C224, 5AC312, 5C031, 5C032, 5C060 w/socket converter from outside source—PLCCs: 85C220, 85C224, 5AC312, 5C060
	PLD-1128	—	V1.46—DIPs: 85C220, 85C224, 5AC312, 5C031, 5C032, 5C060 w/socket converter from outside source—PLCCs: 85C220, 85C224, 5AC312, 5C060
Bytek	135H-U	UNICEL	w/LTA00C—DIPs: 5C031, 5C032, 5C060, 5C090, 5C180, 5AC312, 85C220, 85C224
	145H-AD	—	DIPs: 5C031, 5C032, 5C060, 5C090, 5C180, 5AC312, 85C220, 85C224
Data I/O	UNISITE	SITE 40/48	V3.1—DIPs: 85C220, 85C224, 85C060, 85C508, 5AC312, 5AC324, 5C031, 5C032, 5C060, 5C090 V3.1—PLCCs: 85C220, 85C224, 85C060, 85C508, 5AC312, 5AC324, 5C060, 5C090, 5C180
	Model 2900	—	V1.1—DIPs: 85C220, 85C224, 85C060, 85C508, 5AC312, 5AC324, 5C031, 5C032, 5C060, 5C090 V1.1—PLCCs: 85C220, 85C224, 85C060, 85C508, 5AC312, 5AC324, 5C060, 5C090
	Model 29B	LogicPak V.4	303A-011A (V10)—DIPs: 85C220, 85C224, 85C060, 5AC312, 5C031, 5C032, 5C060 303A-011B (V05)—PLCCs: 85C220, 85C224, 85C060, 5AC312, 5C031, 5C032, 5C060 303A-010 (V03)—DIP & PLCC: 85C090, 5C090
	Model 60A/H	—	DIPs: 5C031, 5C032, 5C060
Digelec	860	—	DIPs: 5C031, 5C032, 5C060, 5C090
		w/8601	PLCCs: 5C060, 5C090, 5C180
Elan	1014	—	DIPs: 5C032, 5C060, 5C090 w/socket converter from outside source—PLCCs: 5C060, 5C090
	5-145	—	DIPs: 5C032, 5C060, 5C090 w/socket converter from outside source—PLCCs: 5C060, 5C090

## PLD Programming Support (Continued)

Vendor	Product	Module	Adaptor/Device/Version/Etc.
Intel	iUP-PC iUP-200A/201A	GUPI Base (both prog.)	GUPI LOGICIID—DIPs: 85C060, 85C090, 5AC312, 5C060, 5C090 w/ADAPT24TO28—PLCCs: 85C060, 5AC312, 5C060 w/ADAPT40TO44—85C090 (PLCC), 5C090 (PLCC) GUPI 20D20J—85C220 (DIP & PLCC), 5C032 (DIP), 5C031 (DIP) GUPI 24D28J—85C224 (DIP & PLCC) GUPI 40D44J—5AC324 (DIP & PLCC) GUPI LOGIC-18—5C180 (PLCC) GUPI LOGIC-18G—5C180 (PGA)
Kontron	EPP-80  MPP-80	UPM/B UPM/C  UPM/B UPM/C	5C031, 5C032, 5C060, 5C090, 5C180 5C031, 5C032, 5C060, 5C090, 5C180 5C031, 5C032, 5C060, 5C090, 5C180 5C031, 5C032, 5C060, 5C090, 5C180
Logical Dev.	ALLPRO	—	V2.1—DIPs: 85C220, 85C224, 85C060, 85C090, 85C508, 85C960, 5AC312, 5AC324, 5C031, 5C032, 5C060, 5C090 w/socket converters from outside source—PLCCs: 85C220, 85C224, 85C060, 85C090, 85C508, 5AC312, 5AC324, 5C060, 5C090, 5C180
Minato	1890A	—	5C031, 5C032, 5C060, 5C090, 5AC312, 85C060, 85C220, 85C508, 85C960
Oliver Adv. Eng.	OMNI-28  OMNI-40  OMNI-64	—  —  —	DIPs: 85C220, 5C031, 5C032, 85C224, 85C060, 5AC312, 5C060 w/OM-S-20 LCC—85C220 (PLCC) w/OM-S-24 LCC—85C224 (PLCC), 85C060 (PLCC), 5C060 (PLCC) All DIPs above + DIPs: 85C090, 5AC324, 5C090, 5C121 w/OM-S-40 LCC—85C090 (PLCC), 5AC324 (PLCC), 5C090 (PLCC) All DIPs above w/OM-S-68 LCC—5C180 (PLCC)
SMS GmbH	Sprint Plus	— w/Pod	V3.2i—DIPs: 5C031, 5C032, 5C060, 5C090, 5AC312 5C180 (PLCC)
Stag	ZL-30 ZL30A ZL33 PPZ System 3000	— — w/30A640 — — —	DIPs: 5AC312, 5C031, 5C032, 5C060 DIPs: 5AC312, 5C031, 5C032, 5C060 5AC312 (PLCC), 5C060 (PLCC), 5C090 (DIP & PLCC) DIPs: 5C031, 5C032, 5C060 DIPs: 5C031, 5C032, 5C060, 5C090 DIPs: 5AC312, 5C031, 5C032, 5C060, 5C090, 5C180
System General	SGUP-85A  TURPRO-1	—  —	V2.0—DIPs: 85C220, 85C224, 85C508, 85C960, 5AC312, 5C031, 5C032, 5C060, 5C090 w/adapters—PLCCs: 85C220, 85C224, 85C508, 85C960, 5AC312, 5C060, 5C090 V1.1—85C220, 85C224, 85C508, 85C090, 5AC312, 5AC324, 5C031, 5C032, 5C060, 5C090, 5C180

## PLD Programming Vendors

The following table lists programming vendors and their products that support Intel PLDs.

Vendor Information	Product	Description
Advin Systems Inc. 1050 E. Duane Ave., Suite L Sunnyvale, CA 94086 (408) 984-8600	Sailor PAL/SA Sailor PAL/SB	Univ. 28 Pins + Expansion Modules Univ. 28 Pins + Expansion Modules Univ. 24-, 28-, 32-, or 40-Pin Programmers + Expansion Modules (PLCC)
BP Microsystems 10681 Haddington, #190 Houston, TX 77043 (800) 225-2102 or (713) 461-9430	PLD-1100 PLD-1128	20/24 Pins + PLCC Socket Converters 28 Pins + PLCC Socket Converter
Bytek Corp. Instrument Systems Division 543 N. W. 77th St. Boca Raton, FL 33487 (800) 523-1565, (407) 994-3520 FAX (407) 994-3615	135H-U 145H-U	Universal MULTIPROGRAMMER® Logic Programmer
Data I/O Corp. 10525 Willows Road, N.E. P.O. Box 97046 Redmond, WA 98073-9746 (800) 247-5700 or (206) 881-6444 FAX (206) 882-1043	UNISITE 40/48 Model 2900 Model 29B Model 60A/H	Univ. 68 Pins—CHIPSITE for PLCC Univ. 40 Pins—DIP and PLCC Univ. 40 Pins—LogicPak for PLDs Univ. 28 Pins—H = Handler Version
Digelec Inc. 20144 Plummer St. Chatsworth, CA 91311 (800) 367-8750 or (818) 701-9677 FAX (818) 701-5040	Uniport	PLD Programmer + PLCC Adaptor
Elan Digital Systems 538 Valley Way Milpitas, CA 95035 (800) 541-3526 or (408) 946-8495 FAX (408) 946-0351	1014 5-145	Univ. + PLCC Socket Converters Programmer + PLCC Socket Converters
Intel Corporation (See Sales Office and Distributor Listings at Back of Handbook.)	iUP-PC iUP-200A/201A	Intel PC-Based + GUPI Adaptors Intel Standalone + GUPI Adaptors
Kontron 244 Sobrante Way Sunnyvale, CA 94086 (800) 227-8834 or (408) 733-0272 FAX (408) 733-0378	EPP-80 MPP-80	Univ. + UPM Modules Univ. + UPM Modules
Logical Devices, Inc. 1201 N.W. 65th Place Ft. Lauderdale, FL 33309 (800) 331-7766 or (305) 974-0967 FAX (305) 974-8531	ALLPRO ALLPRO 88	Univ. 32 or 40 Pins + PLCC Socket Converters Univ 88-Pin + PLCC Socket Converters



The following table lists programming vendors and their products that support Intel PLDs.

Vendor Information	Product	Description
Oliver Advanced Engineering 320 Arden St. Glendale, CA 91203 (800) 828-0080 or (818) 240-0080 FAX (818) 240-6131	OMNI-64 OMNI-40 OMNI-28	Univ. 64 Pins Univ. 40 Pins Univ. 28 Pins
SMS GmbH Im Morgenthal 13 D-8994 HERGATZ Germany	Sprint Plus	Univ. 28 Pin + Expansion Pods
Stag Microsystems, Inc. 1600 Wyatt Drive, Suite 3 Santa Clara, CA 95054 (800) 227-8836 or (408) 988-1118 FAX (408) 988-1232	ZL-30 ZL-30A ZL33 PPZ	Logic 28 Pin + PLCC Adaptor Logic 28 Pin + PLCC Adaptor Gang 24 Pin Univ. 28 Pin + PLCC
System General Corp. 244 S. Hillview Dr. Milpitas, CA 95035 (408) 263-6667 FAX (408) 262-9227	SGUP-85A	Univ. 68 Pins
UEC/PROMAC 1095 Market Street Suite 701 San Francisco, CA (415) 255-9393 FAX (415) 255-9392	Model 11A	5C031, 5C032, 5C060, 5C090, 5C180, 5AC312

Univ. = Universal Programmer

## PLD Software Support

Vendor	Product	Version/Devices Supported
Data I/O	ABEL	V4.0—85C220, 85C224, 85C060, 85C090, 85C508, 5AC312, 5AC324, 5C031, 5C032, 5C060, 5C090, 5C180
	GATES PLDTest	V5.0—Contact Vendor for Supported Devices V1.3—Contact Vendor for Supported Devices
Hewlett-Packard	HP PLD Design System	V3.4—5C031, 5C032, 5C060, 5C090, 5C180
Intel Corp.	PLDshell Plus	V3.0—iFX780, iPLD610, iPLD910, iPLD22V10, 85C060, 85C090, 85C22V10, 85C508, 5AC312, 5AC324, 5C031, 5C032, 5C060, 5C090, 5C180
ISDATA	Log/IC	V3.3—85C220, 85C224, 85C508, 5AC312, 5C031, 5C032, 5C060, 5C090, 5C180
Logic Automation	Model Library	5AC312, 5AC324, 5C031, 5C032, 5C060, 5C090, 5C180
Logical Devices	CUPL	V3.3—85C220, 85C224, 85C060, 85C090, 85C508, 5AC312, 5AC324, 5C031, 5C032, 5C060, 5C090, 5C180
MINC Inc.	PLD Designer	V1.0—5C031, 5C032, 5C060, 5C090 V1.8—5C180, 85C220 V2.0—85C508 V2.1—5AC312, 85C224, 85C060, 85C090
OrCAD Systems	SDT III  PLD VST	V4.04—85C220, 85C224, 85C060, 85C090, 85C508, 5AC312, 5AC324, 5C031, 5C032, 5C060, 5C090, 5C180 V4.02—Contact Vendor for Supported Devices V4.02—Contact Vendor for Supported Devices
Silicon West	Simulation Models	85C220, 85C224, 85C508, 5C032, 5C060, 5C090
Viewlogic, Inc.	Workview CAE	V4.0—85C220, 85C224, 85C060, 85C090, 85C508, 85C960, 5AC312, 5AC324, 5C031, 5C032, 5C060, 5C090, 5C180

## PLD Software Vendors

The following table lists programming vendors and their products that support Intel PLDs.

Vendor Information	Product	Description
Data I/O Corp. 10525 Willows Road, N.E. P.O. Box 97046 Redmond, WA 98073-9746 (206) 881-6444	ABEL GATES PLDTest	Logic Compiler S/W Logic Synthesis S/W PLD Test S/W
Hewlett-Packard Company Customer Information Center (1-800) 752-0900	HP PLD Design System	Compiler/Synthesis S/W
Intel Corporation (See Sales Office and Distributor Listings at Back of Handbook.)	PLDshell Plus	Compiler/Simulation S/W
ISDATA GmbH Haid-und-New-Straße7 D-7500 Karlsruhe West Germany	LOG/iC (Outside U.S. & Canada)	Compiler/Partitioner S/W
ISDATA, Inc. 800 Airport Rd. Monterey, CA 93940 (408) 373-3607	LOG/iC (U.S. & Canada)	Compiler/Partitioner S/W
Logical Devices, Inc. 1201 N.W. 65th Place Ft. Lauderdale, FL 33309 (800) 331-7766 or (305) 974-0967 FAX (305) 974-8531	CUPL	Compiler S/W
MINC, Inc. 6755 Earl Drive Colorado Springs, CO 80918 (719) 590-1155	PLDesigner-XL	Compiler/Partitioner S/W
OrCAD Systems Corp. 3175 N.W. Alcock Dr. Hillsboro, OR 97124-7135 (503) 690-9881	OrCAD/SDT III OrCAD/VST	Schematic S/W Simulation S/W
Silicon West 5150 E. Pacific Coast Highway Suite 320 Long Beach, CA 90804 (213) 494-4127	Simulation Models	See "Software Support" for Devices Supported; Contact Vendor for Platforms/Simulators and Devices Supported
Viewlogic, Inc. 313 Boston Post Road West Marlboro, MA 01752 (508) 480-0881	Intel PLD Kit	Intel PLD Design Library for Workview— Schematic/Simulation S/W

## PAL\*/GAL\* to Intel PLD Replacement

Already in wide use throughout the electronics industry are numerous different Programmable Logic Devices. Most common PALs and GALs can be replaced or upgraded with the following Intel PLDs:

### 85C220

The 85C220 is a direct, drop-in replacement for most 20-pin PALs/GALs, although some PALs have an incompatible architecture. The 85C220 runs at 80 MHz with external feedback. 85C220 to 20V8 cross programming is now available through the cross programming menu on Data IO's Unisite, Autosite 2900 and 3900 programmers.

### 85C224

The 85C224 is a direct, drop-in replacement for most 24-pin PALs/GALs, although some PALs have an incompatible architecture. The 85C224 runs at 80 MHz with external feedback. 85C220 to 20V8 cross programming is now available through Data IO's Unisite, Autosite 2900 and 3900 programmers.

### iPLD22V10

The iPLD22V10 is 100% pin-, function- and JEDEC-compatible with industry standard 22V10 devices. JEDEC files developed for 22V10 devices can be used "as is" to program the iPLD22V10.

### 85C220 As a 20-Pin PAL Replacement

100% Compatible	
10H8, -2	16R6A
12H6, -2	16R4A
14H4, -2	16L8A
16H2, -2	16RP6A
10L8, -2	16RP4A
12L6, -2	16P8A
16L8, A-2, A-4	16R8A
16R4, A-2, A-4	16RP8A
14L4, -2	16V8A
16L2, -2	18P8
16R8, A-2, A-4	18V8
16R6, A-2, A-4	
16P8, -2	
16RP8, -2	
16RP6, -2	
16RP4, -2	
16V8	

### 85C224 As a 24-Pin PAL Replacement

100% Compatible	
14L8	20L8A
16L6	20R8A
18L4	20R6A
20L2	20R4A
20L8	20V8
20R8	
20R6	
20R4	

\*PAL is a registered trademark of Advanced Micro Devices.

\*GAL is a registered trademark of Lattice Semiconductor, Incorporated.



## Intel PLD Feature Comparison

	85C220		iPLD610/		iPLD910/			
	5C032	85C224	5C060	5C090	iPLD22V10	5C180	5AC312	5AC324
<b>INPUTS</b>								
Dedicated	10	14	4	12	12	12	10	12
Maximum	18	22	20	36	22	60	22	36
Input Latches/Registers							Y	Y
<b>I/O</b>								
Number	8	8	16	24	10	48	12	24
Tri-State	Y	Y	Y	Y	Y	Y	Y	Y
Programmable	Y	Y	Y	Y	Y	Y	Y	Y
Polarity								
Dual-Feedback							Y	Y
<b>MACROCELLS</b>								
	8	8	16	24	10	48	12	24
<b>REGISTERS</b>								
Number	8	8	16	24	10	48	12	24
Types	D	D	D/T/ RS/JK	D/T/ RS/JK	D	D/T/ RS/JK	D/T/ RS/JK	D/T/ RS/JK
By-Pass	Y	Y	Y	Y	Y	Y	Y	Y
Reset to 0	Y	Y	Y	Y		Y	Y	Y
Preset to 1							Y	Y
<b>PRODUCT TERMS</b>								
Number	72	72	160	240	92	480	200	394
Allocation							Y	Y
<b>LOCAL/GLOBAL BUSES</b>								
						Y		
<b>CLOCKS</b>								
Asynchronous	1	1	2	2	1	4	2	2
Clocking			Y	Y		Y	Y	Y
<b>SECURITY BIT</b>								
	Y	Y	Y	Y	Y	Y	Y	Y
<b>TURBO BIT (LOW POWER)</b>								
	Y	Y	Y	Y		Y	Y	Y

## Industrial Temperature Devices

Intel offers industrial temperature PLDs. Unless otherwise specified, commercial temperature specifications apply to industrial temperature devices. The table below lists the industrial temperature devices.

Device	Package	Speeds	Order as
5C032	D, P	35, 40	TD5C032-35 TD5C032-40 TP5C032-35 TP5C032-40
5C060	D, P, N	45, 55	TD5C060-55 TP5C060-55 TN5C060-55
5C090	D, P, N	60	TD5C090-60 TP5C090-60 TN5C090-60
5C180	N	90	TN5C180-90
5AC312	D, N	30	TN5AC312-30
5AC324	N	30	TN5AC324-30
85C220	D, N	66	TD85C220-66 TN85C220-66
85C060	D, N	15, 25	TD85C060-15 TD85C060-25 TN85C060-15 TN85C060-25
85C090	D, N	20, 25	TD85C090-20 TD85C090-25 TN85C090-20 TN85C090-25

5

## Military Devices

Intel offers the following military PLDs.

Device	Package	Speeds	Order as
5C060	D	55	MD5C060-55
5C090	D	60	MD5C090-60
5C180	A, K	90	MG5C180-90 MQ5C180-90
5AC312	D	30, 35	MD5AC312-30 ME5AC312-35
85C220	D	12, 15, 50, 66	MD85C220-12 MD85C220-15 MD85C220-50 MD85C220-66

For detailed information on military devices, refer to the *Military Products Handbook*, Order Number 210461.

## Tape and Reel Packaging

Intel offers tape and reel packaging of PLCC devices in North America. Please contact your local Intel sales representative for details.

## Pre-Programmed Devices

Intel has the capability of providing pre-programmed and specially tested devices in North America and Japan. Please contact your local Intel sales representative for details.

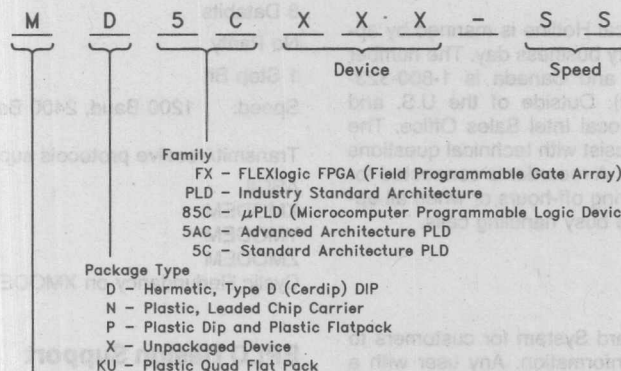
## Register Preload

Most Intel  $\mu$ PLD macrocell registers can be preloaded with any pattern to allow testing of all possible logic states. Information on register preload for test purposes is available from Intel.

88C080	D, N	30, 35	T188C080-25 T188C080-30 T188C080-35
88C080	D, N	15, 25	T188C080-15 T188C080-25
88C250	D, N	85	T188C250-80 T188C250-90
AC824	N	30	T188AC824-30
2AC812	D, N	30	T188AC812-30

# Ordering Information

Intel PLDs are identified as follows:



000273-1

- A — Indicates automotive operating temperature range ( $-40^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$ )
- J — Indicates a JAN qualified device, but is for internal identification purposes only. All JAN devices must be ordered by M38510 part number. (Example: M38510/42001 BQB), and will be marked in accordance with MIL-M-38510 specifications.
- L — Indicates extended operating temperature range ( $-40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$ ) express product with 160 + 8 hrs. dynamic burn-in.
- \*M — Indicates military operating temperature range ( $-55^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$ )
- Q — Indicates commercial temperature range ( $0^{\circ}\text{C}$  to  $+70^{\circ}\text{C}$ ) express product with 160 + 8 hrs. dynamic burn-in.
- T — Indicates extended temperature range ( $-40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$ ) express product without burn-in.
- No letter indicates commercial temperature range ( $0^{\circ}\text{C}$  to  $+70^{\circ}\text{C}$ ) without burn-in.

Examples:

QD5C060-45 Commercial with burn-in, ceramic Dip, 060 (600 gate) device, 45 nanosecond.

\*On military temperature devices, B suffix indicates MIL-STD-883C level B processing.



## EPLD Customer Support

### Hotline

The Intel EPLD Technical Hotline is manned by application personnel every business day. The number for the United States and Canada is 1-800-323-EPLD (1-800-323-3753). Outside of the U.S. and Canada, contact your local Intel Sales Office. The Hotline is provided to assist with technical questions concerning Intel EPLDs. A recorder is connected for receiving messages during off-hours or when all applications personnel are busy handling calls.

### BBS

Intel has a Bulletin Board System for customers to electronically transfer information. Any user with a modem can log onto the system. The current number is (916) 985-2308. If your communication software supports file transfers, you can receive utilities, software updates, and the latest information on EPLDs via the Bulletin Board.

Data format for the BBS is as follows:

8 Databits

No Parity

1 Stop Bit

Speed: 1200 Baud, 2400 Baud, or 9600 Baud

Transmit/receive protocols supported are:

ASCII

XMODEM

YMODEM

ZMODEM

Cyclic Redundancy on XMODEM

### EPLD Design Support

Intel has hardware designers who can help you with your EPLD designs. For more information on design assistance, contact your local Intel field sales office.